It's all about Leopard

MacTech Magazine
November, 2007

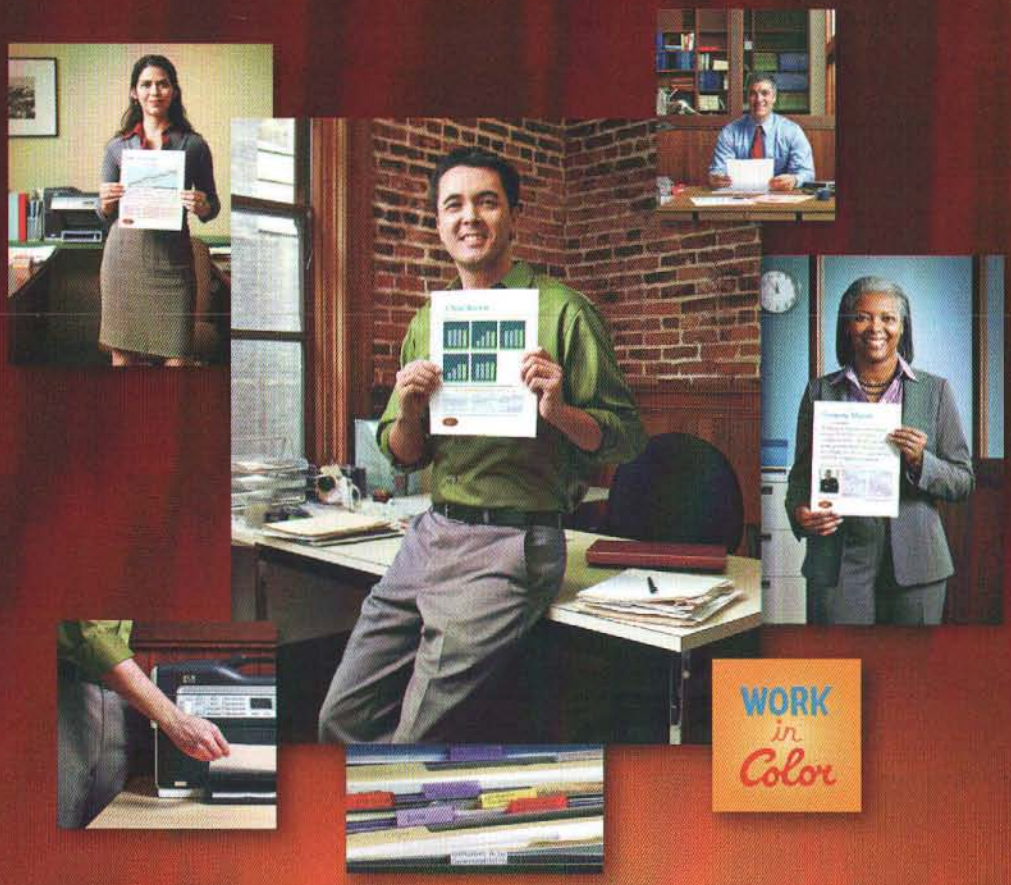
MACTECH®

The Journal of Macintosh Technology

MACTECH.COM

$8.95 US, $12.95 Canada

ISSN 1067-8360 Printed in U.S.A.



UP TO 25% LOWER COST PER PAGE THAN LASER PRINTERS.* HP OFFICEJET PRO ALL-IN-ONES STARTING AT $299.

They fax, scan, copy and deliver laser-quality printing—at an impressively low cost per page. Individual ink cartridges save money too, because you only replace what runs out. HP, *PC Magazine*'s Readers' Choice for service and reliability for 15 years running.

BEST
HP Officejet Pro L7780 AiO.
Plus wireless networking.
$499.

BETTER
HP Officejet Pro L7680 AiO.
Plus two-sided printing
$399.

GOOD
HP Officejet Pro L7580 AiO.
Up to 35 pages a minute.
$299.

To learn more, visit hp.com/go/doesitall. Call 1-800-888-3119. Or visit your local reseller or retailer.

*Cost per page (CPP) laser supplies comparisons based on published manufacturer specifications of the highest capacity cartridges available for color laser all-in-ones under $1,000 and mono laser all-in-ones under $500, as reported by Current Analysis, Inc. as of July 2006. Officejet Pro L7500/L7600/L7700 AiO CPP based on HP 88 Large Ink Cartridges (not included, sold separately) estimated street price and published yield. Results may vary. Estimated U.S. retail price. Actual price may vary. Simulated images.

HASP, the #1 Choice of Software Developers Worldwide*

*Source: Frost & Sullivan #N1AF-70, IDC Bulletin #34452

Finally, put a

STOP

to unauthorized software use.

HASP® HL

REINVENTING SOFTWARE PROTECTION & LICENSING

Follow the signs to HASP HL – the market's leading software DRM solution

HASP HL provides the highest level of security for your software. Featuring easy-to-use tools, HASP HL automatically protects your software and enables new and innovative licensing options.

- ***Strong copy protection:*** *Powerful 128-bit AES encryption provides a strong locking mechanism that ensures you get paid for every copy of your software in use.*
- ***Unmatched intellectual property protection:*** *Automatically protects Windows executables, DLLs, Mac applications, and offers class-level obfuscation for .NET framework 2.0 applications. Multiple encryption layers ensure your valuable IP remains unreachable and your R&D investment is protected.*
- ***Secure & flexible licensing:*** *HASP HL enables a variety of secure and flexible licensing models and ensures end-users comply with software licensing terms.*

See for yourself why IDC ranks HASP #1.

Request a Software Developer Kit today at www.Aladdin.com/StopPiracy_MT

Aladdin

SECURING THE GLOBAL VILLAGE

• NORTH AMERICA: 1-800-562-2543, 847-818-3800
• UK • GERMANY • ISRAEL • BENELUX • FRANCE • SPAIN • ITALY • BRAZIL • INDIA • CHINA • JAPAN

www.Aladdin.com/HASP

The ALL-N

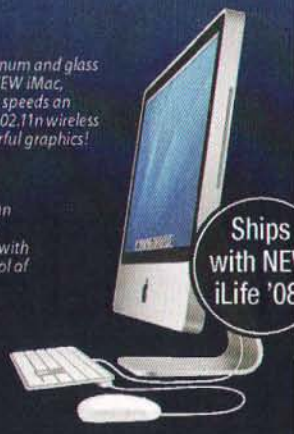
The NEW Apple® iMac® now boasts faster speeds, your choice of Intel Core 2 Duo or Intel Core 2 Extreme processing, a bigger hard drive, new graphics and more! And it all comes in a new sleek and stylish aluminum casing that turns your desktop into the centerpiece of your home or office! Call MacMall today and discover this union of elegance and functionality for yourself!

New!

MacMall Exclusive Deals!
Get up to $200 cash back!*
Parallels Desktop 3.0 only $99!*
HP or Epson Printer FREE!*
Laptop carrying case FREE!*
See below for details.

Inside the aluminum and glass design of your NEW iMac, you'll find faster speeds an iSight camera, 802.11n wireless and more powerful graphics!

The NEW iMac includes an ultra-thin keyboard with two USB 2.0 ports along with keys for one-touch control of special Mac functions!



Ships with NE iLife '08

Apple MacBook Pro™

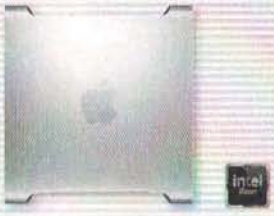
- NEW! Intel Core 2 Duo speeds up to 2.4GHz
- NEW! NVIDIA video graphics
- NEW! Mercury-free 15" LCD
- Built-in iSight™, wireless & Gigabit Ethernet

starting at ~~$1994~~ -$150 = **$1844!*** #7254324

Apple MacBook™

- 13" glossy widescreen display
- Intel Core 2 Duo processing up to 2.16GHz
- AirPort Extreme 802.11n wireless
- Built-in iSight™ and Gigabit Ethernet

starting at ~~$1094~~ -$75 = **$1019!*** #7248171



8-Core and 4-Core Mac® Pro

- Two Dual-Core Intel Xeon processors or two Quad-Core Intel Xeon processors
- Incredible storage expansion up to 3TB
- 16X SuperDrive™ (DVD+R DL/DVD±RW/CD-RW)

starting at ~~$2194~~ -$100 = **$2094!*** #7098933

Faster wireless! Advanced storage! Cool peripherals!

Apple iLife '08
#7288385
New!
only $69.98!
#7288386 Apple iLife '08 Family Pack ~~$89~~

VMware Fusion for Mac
#7288411
~~$64.98~~ $44.98!
After $20 mfr. mail-in rebate. Price before rebate is $64.98. Ends 12/31/07†

500GB d2 Quadra HD eSATA, FW800/400 and USB 2.0
#7185563
only $189.99!

Apple iWork™ '08
#7288383
only $69.98!
#7288384 Apple iWork '08 Family Pack ~~$89~~

V470 Cordless Optical Bluetooth Notebook Mouse
#7297600
only $53.99!

320GB d2 Quadra HD eSATA, FW800/400 and USB 2.0
#7244917
was ~~$174.99~~
now $159.99!

Top tools for media pros!

QuarkXPress 7
#7175856
upgrade only $239.99!

Apple Final Cut® Studio
#723621
New!
upgrade only $473.99
#7236099 Final Cut Studio Full Package ~~$1199~~

Apple AirPort® Extreme Base Station
802.11a/b/g/n
#7288377
New!
only $178.99!

Apple 23" Cinema HD Display
#459837
was ~~$994~~
now $894!

WE'LL BEAT ANY PRICE ON ANY APPLE BRAND PRODUCTS or competitive promotion on any Apple computer. We may require proof of dealer's pricing. Competitor must be an Apple Authorized Reseller. Product must be in stock at the competition. Advertised items must be identical to the one offered by MacMall and must be a new item factory sealed box. We reserve the right to limit quantity ordered. This offer does not apply to special, bonus or free offers, grand openings, special purchases or special buys, manufacturers rebates, closeouts, clearances or to dealers' one-of-a-kind or other limited quantity offers. Nor does it apply to advertising errors made by any authorized dealer. Not c to dealers. Void where prohibited. Sorry, offer not valid in conjunction with other rebate offers. On multiple item quotes: "We'll Beat Any Price" applies to the price of the entire order, not individual line items. Offer applies to pre-taxed prices only. Offer valid only at time of purchase, not after the fact. ■ UP TO $200 CASH BACK OFFER-Get up to $200 Cash via MacMall mail-in rebate with purchase of select computer models. Ends 11/8/07. ■ LAPTOP CARRYING CASE OFFER-Get a laptop carrying case FREE after $29.99 mfr. mail-in rebate. Price before rebate is $29.99. Ends 11/8/07. ■ PARALLELS DESKTOP OFFER-Get Parallels Desktop 3.0 for Mac #7257416 for only $9.99 after $10 mfr. and $60 Mac mail-in rebates with purchase of any new Intel-based Apple CPU. Price before rebates is $79.99. Ends 11/8/07. ■ FREE PRINTER OFFER-Get an HP PhotoSmart C3180 All-in-One FREE after $30 MacMall instant rebate and after $29.99 MacMall mail-in and $50 mfr. mail-in rebates with purchase of any Apple computer. Price before rebates is $99.99. Of an Epson Stylus Photo RC90 Printer FREE after $19.99 MacMall mail-in and $70 mfr. mail-in rebates with purchase of any Apple computer. Price before rebate is $89.98. Offers may not be combined. Ends 11/8/07. ■ FREE ENGRAVING OFFER-Get select iPod, iPod touch, iPod classic, iPod nano and iPod shuffle models engraved for FREE when you purch them through MacMall. No rebate required. There is a $9.99 charge for select iPods. Call or visit www.macmall.com/ipod to place your order. ■ FREE SHIPPING OFFER-Free UPS ground shipping, after mail-in-rebate, for most orders $125 or more (not including taxes and shipping costs). Limit one rebate per mailing address. Total rebate amount not to exceed $40 per shipping address. Valid only if the products is purchased at the price listed on the Web on the day the order is placed. Rebate must be postmarked within 20 days of shipping invoice date. Offer good online or through MacMall's call center at 1-800-622-6255. Offer valid for all orders placed before 11:59 p.m. PDT on 11-8-07. Applies to ground orders only. ■ FREE FM TRANSMITTER OFFER-After $19.99 MacMall mail-in rebate. Price before rebate is $19.99. Ends 11-8-07. ■ ALL OFFERS VALID ONLY WHILE SUPPLIES LAST. ■ †For rebate terms and conditions, please visit our Web site and enter the applicable part number. Download rebate coupons at www.macmall.com/rebates.

N iMac.

NEW! Intel Core 2 Duo processor at up to 2.4GHz
or Intel Core 2 Extreme processor at 2.8GHz
NEW! Up to 320GB Serial ATA hard drive
NEW! Glossy widescreen display
NEW! ATI Radeon HD video graphics
NEW! Aluminum casing and ultra-thin keyboard
iSight™, 802.11n wireless and Gigabit Ethernet

~~1194~~ -$50 *(mail-in rebate)*

Mac starting at
$1144! *Finance for $36/mo*

EW 20" Apple iMac®—2GHz Intel Core 2 Duo
'288171 only ~~$1194~~ -$50 = $1144!*
EW 20" Apple iMac®—2.4GHz Intel Core 2 Duo
'288181 only ~~$1494~~ -$75 = $1419!*
EW 24" Apple iMac®— 2.4GHz Intel Core 2 Duo
'288182 only ~~$1794~~ -$100 = $1694!*
EW 24" Apple iMac®— 2.8GHz Intel Core 2 Extreme
'288393 only ~~$2294~~ -$100 = $2194!*

EW Mac mini—up to 39% faster!
NEW! Intel Core 2 Duo processor up to 2GHz
NEW! 1GB DDR2 SDRAM; exp. to 2GB
NEW! Up to 120GB Serial ATA hard drive
iSight™, 802.11g wireless & Gigabit Ethernet

ting at ~~$594~~ -$25 = **$569!*** #7288184

New Adobe Creative Suite 3!

NEW Master Collection CS3!
only $2450! #7228222

NEW Adobe Creative Suite 3!
Upgrades from $393.99!

MacMall is your source for LCD and plasma widescreen TVs!

Panasonic SAMSUNG ViewSonic OLEVIA SHARP

NEW iPod touch!

Music! Wi-Fi Web! Movies! And more!

iPod touch features the same revolutionary multi-touch interface as the iPhone! Browse the Web wirelessly and show off your videos, music and photos in colorful widescreen glory!

- Holds up to 20,000 songs (16GB model)
- View movies, photos and YouTube videos
- Wi-Fi Web browsing ■ Multi-touch interface
- 3.5" widescreen rotating display

8GB iPod touch only
$294! #7297636
#7297637 iPod touch 16GB $394

Authorized Reseller

FREE Engraving, FM Transmitter and Shipping!

New!

NEW iPod nano!

- NEW! Holds up to 5 hours of TV shows, music videos, movies and Podcasts
- NEW! 320 x 240 display that's 65% brighter

4GB Silver only **$144!** #7297625

#7297626	NEW iPod nano 8GB Silver	$194
#7297628	NEW iPod nano 8GB Blue	$194
#7297630	NEW iPod nano 8GB Green	$194
#7297627	NEW iPod nano 8GB Black	$194

FREE FM Transmitter and Engraving!

New!

NEW iPod shuffle—remixed colors!

- NEW! Four soothing colors
- Holds up to 240 high-fidelity songs
- Incredibly compact clip-on design

1GB Blue only **$78.99!** #7297622

#7297618	NEW 1GB iPod shuffle Silver	$78.99
#7297623	NEW 1GB iPod shuffle Green	$78.99
#7297624	NEW 1GB iPod shuffle Purple	$78.99

NEW iPod classic!

- NEW! All-metal design
- NEW! Cool games
- Holds up to 40,000 songs (160GB model)
- 2.5" color LCD

FREE Engraving, FM Transmitter and Shipping!

80GB Silver only **$244!** #7297631
#7297635 iPod classic 160GB Black $344

Hot NEW iPhone accessories!

Visit www.macmall.com today for the best selection anywhere!

MacMall is the #1 Apple Direct Reseller!

MacMall®
1-800-328-2794
Visit us at macmall.com
Source Code: MACTECH

Premium

Small Business Management & Accounting

Software

Mind Your Own Business. Smarter.

800.322.MYOB (6962)

www.myob-us.com

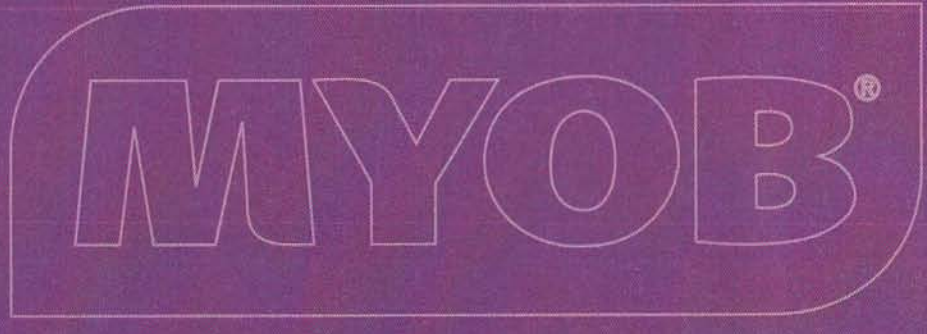


TABLE OF CONTENTS

ARTICLES & DEPARTMENTS

From the Editor

November brings Autumn in full swing with leaves on the ground, sweaters, and sitting by the glowing embers of our LCD screens. Leopard has been released and while many of us have had access to beta versions, we didn't always have the best *documentation*. Enter this month's MacTech. Starting with this issue, we're dedicated to getting you the information you need about Leopard.

The more we've dug into Leopard, the more we're finding that is revolutionary than evolutionary. Regular end-users will notice many of the nice little touches that show up in 10.5, but the real action is under the hood. When I initially heard developers speak about making their next releases Leopard-only, I thought that may be a short-sighted move. However, Leopard does so much for the developer and tech-type, I can understand that thinking now. End users *will* benefit from the parts of the OS that they never touch themselves by developers taking advantage of them!

To lead off our Leopard coverage, *do not miss* **Greg Miller's** article on **DTrace**. If you haven't noticed yet, gone are ktrace and some of our other tried-and-true debugging tools. As the biggest beneficiary to the open-sourcing of Sun's Solaris, Apple puts dtrace into 10.5. Now, dtrace is incredibly powerful, but isn't quite as straight-forward as some other tools. Greg has written the clearest introduction to dtrace that I've seen.

Ben Greisler talks to sysadmins and points out the **5 technologies in Leopard** that they should be checking out right away. Separately, Ben has written an article on the inclusion of **RADUIS** support in Leopard – a very welcome addition, praised by system administrators familiar with the benefits it brings.

Dave Dribin takes the Leopard side-street in this month's **Road to Code** to talk about **what's new for developers**. Dave points out new features in XCode, Objective C 2.0, various frameworks and more.

Philip Rinehart and MacEnterprise get you started with the radical **changes in directory services**. Did you know that the venerable NetInfo is gone in Leopard? Gone! Directory Services play a much larger role in Leopard than ever before.

Well, we couldn't possibly cover *all* of Leopard in a single issue, and there are still plenty of non-Leopard-specific topics to discuss. One of my favorites this month is an article by **Joe Froehlich** that teaches us about a device that **converts a VGA signal to one that can be captured** – independent of the OS – on the USB.

Ever need a little more storage? Of course, we all do! However, sometimes, you're looking for something more than a single Firewire drive, but less than a fully-loaded XServe RAID. **Robert Staehle** shows us how to use **FreeNAS**, an Open Source FreeBSD-based NAS system that you can use at home, or in a business setting.

Back in Leopard land, don't forget that not only is the OS overhauled, but the built-in apps are, too. Mail.app, Safari and...Terminal?!? Yes! For you command-line people, Terminal.app 2 has some really nice features, and you can learn about them in this month's **Mac in The Shell**.

Last, but not least, is this month's **MacTech Spotlight**, featuring **Kent Sutherland**. Kent is currently a student at the Rochester Institute of Technology and has already graced us with code that we can't live without (ok, well, some of us). He's most well know for Chax, an Input Manager bundle that adds features to iChat, but he also....well...go read the feature!

Enjoy Leopard, dig in to our coverage here, and dig in on your own. If you find something that just isn't documented anywhere else, let us know! Send your thoughts to feedback@mactech.com. Hope to be hearing from you!

Edward Marczak,
Executive Editor

Communicate With Us

Department E-Mails

Orders, Circulation, & Customer Service
cust_service@mactech.com

Press Releases
press_releases@mactech.com

Ad Sales
adsales@mactech.com

Editorial
editorial@mactech.com
(Authors only, no pr)

Accounting
accounting@mactech.com

Marketing
marketing@mactech.com

General
info@mactech.com

Web Site
http://www.mactech.com

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail?**

If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

If you would like a subscription or need customer service, feel free to contact MacTech Magazine Customer Service at 877-MACTECH

We love to hear from you! Please feel free to contact us with any suggestions or questions at any time.

Write to letters@mactech.com or editorial@mactech.com as appropriate.

MACTECH®
The Journal of Macintosh Technology

A publication of XPLAIN CORPORATION

Canada Post: Publications Mail Agreement #41513541
Canada Returns to be sent to: Bleuchip International, P.O. Box 25542, London, ON N6C 6B2

MacTech Magazine (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 850-P Hampshire Road, Westlake Village, CA 91361-2800. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are $47.00 per year. Canadian subscriptions are $59.00 per year. All other international subscriptions are $97.00 per year. Domestic source code disk subscriptions are $77 per year. All international disk subscriptions are $97.00 a year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

POSTMASTER: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

5 Leopard Features To Make Your Sys Admin Happier

Here are a few features that will make your life as a systems administrator easier

by Ben Greisler

Life as a Sys Admin

Being a systems administrator is hectic and stressful. Lucky for us sometimes we get a bit of help to make life a tad easier. With the introduction of Leopard, we have been given some tools that will help out just the little bit that makes a difference to our daily work.

Here are 5 features that are included in Leopard that while aren't earth shaking, will just help out. Some of these features have been available using third party software or simply by some home brew scripting, but they are now built-in and this means Apple has been listening.

Terminal 2

The two new features in Terminal 2 that I see as the most useful are tabs and window groups. Tabs fall into the "It's about time" category. As handy as tabs are in a browser, they are just as handy in Terminal.

After spending time setting up terminal windows just exactly the way that we want, it is now nice to be able to save all that work in multiple windows as window groups. A simple click brings those windows back to us.

Check out Ed Marczak's "Mac In The Shell" column this month on Terminal 2 in this issue for more information.

Screen Sharing

Sure you say, "I already have ARD. Why do I need Leopard's screen sharing?" The way that I see it, you are wandering around the office floor working with users and you need to check something else on another machine. Bingo! You just fire off screen sharing and off you go. It is as simple as clicking on the Share Screen button in a Finder window. This is a convenience feature that I am sure sys admin's will find great use for as time goes on. Screen sharing is based on VNC so plan accordingly.

Fig. 1. Screen sharing is now a standard OSX feature

Instruments

Instruments is billed as a code-debugging tool with the power of Dtrace, but I see it as much more. Designed to allow a developer to view various aspects of the system while an application is running, this is just what a sys admin needs when troubleshooting issues.

Instruments is part of the Developer Tools package and lives in the Applications folder within Developer folder. It has

WireTap Studio

A revolutionary new way to record, edit, and manage all of your audio.

AMBROSIA®
SOFTWARE INC

http://www.AmbrosiaSW.com/mactech

WireTap Studio requires Mac OS X 10.4 or later. Ambrosia Software, Inc., and the Ambrosia Software logo are registered trademarks of Ambrosia Software, Inc.

Mac Universal

an interface that will be familiar to any Mac user and visualizes what happens when during the use of an application.

Please see Greg Miller's DTrace article (p.66) and Dave Dribin's "Road to Code (p.28)" column for more information.

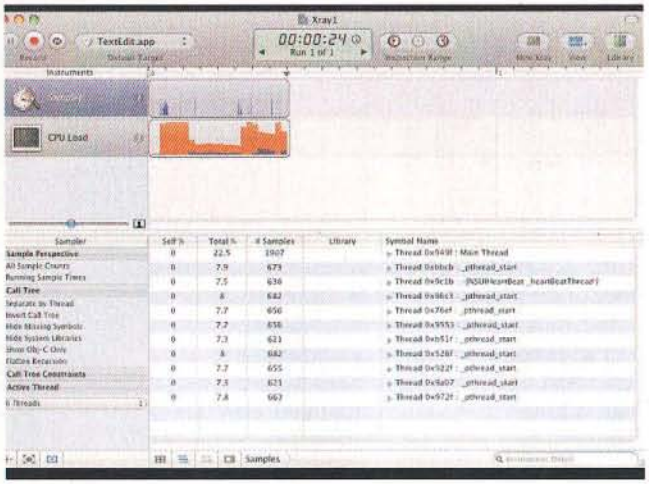


Fig. 2. Instruments allows powerful inspection of applications while they run

Spaces

Another member of "It's about time" features is Spaces. A window manager similar to those available on any standard Linux install, you get to associate applications with virtual desktops. I don't know about you, but I never thought that Expose was a real answer to screen real estate issues, so I am happy to have Spaces as a standard part of the OS.

Life becomes more organized when you can dedicate a screen to a specific purpose. Combine Spaces with a tabbed terminal and you get a very manageable work environment. Combine that with a pair of 30 inch Cinema displays and you have quite the effective administration system.

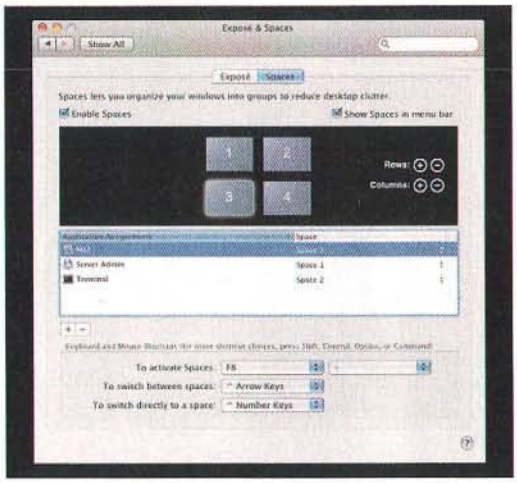


Fig. 3. Configuring Spaces to give each app its own place to live

Server Admin Notifications

While we don't have as comprehensive notifications as I would like, the one simple notification for disk space is a great

download your FREE TRIAL TODAY

MacScan

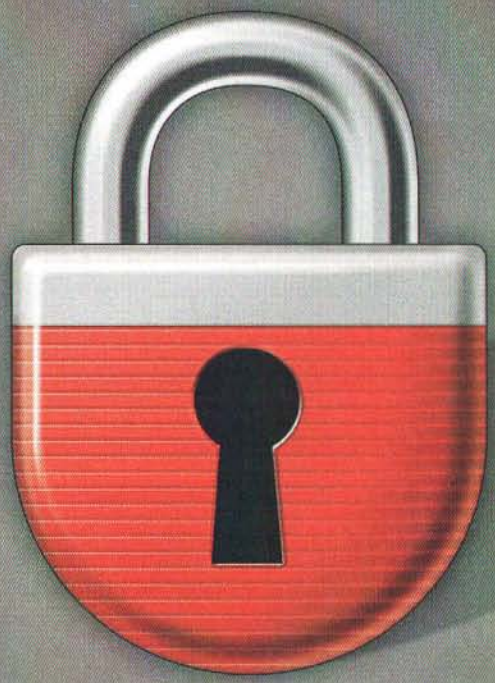
DETECTS, ISOLATES & REMOVES SPYWARE

http://macscan.securemac.com

Mac Spyware Protection

Protect your security and privacy with MacScan, the premier Anti-Spyware for Mac OS X

Spyware exists--and it's a growing threat to the online community. As the number of Mac users increases, so do the threats to your personal information. MacScan delivers peace of mind by auditing your computer for spyware and cleaning up clutter from casual Internet browsing to protect privacy and security. MacScan's Cookie Blacklist allows you to safely scan and remove known tracking cookies from popular web browsers.



OVER 8000 BLACKLISTED COOKIES IN THE DATABASE

AUTOMATICALLY UPDATES SPYWARE DEFINITIONS

SUPPORTS 9 BROWSERS INCLUDING FIREFOX, SAFARI, AND NETSCAPE

Download your free trial today!
http://macscan.securemac.com

Now available at Office DEPOT COMPUSA TEKSERVE MICRO CENTER

Mac Universal

boon to those situation where wider ranging monitoring isn't in place. Sys admins/school librarians who are in charge of OSX servers will love this simple feature that will avoid filled drives that are all too common.

This feature is part of Server Admin in the Notifications tab within Settings. It is limited to sending an email when the free space falls under a defined amount and at this point it doesn't allow you to specify which drives to monitor, but it is a step in the right direction. The File Sharing pane in Server Admin makes a quick job of scanning disk space with the bar graphs and I hope they allow finer grained monitoring in the future. There is also a Server Admin widget for those who find Widgets useful.

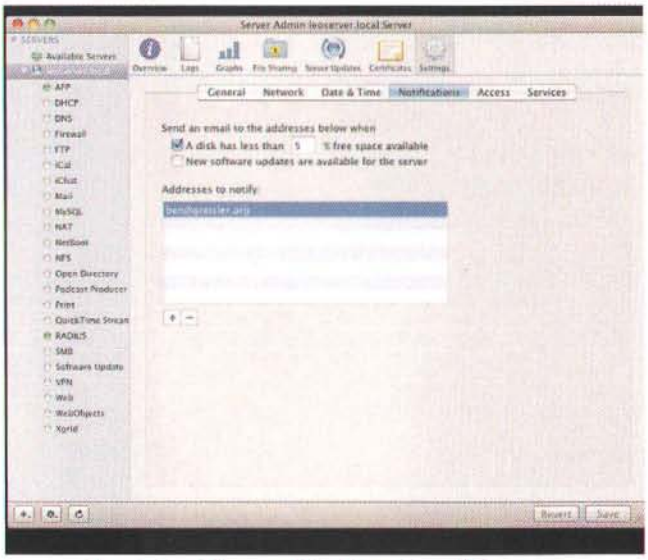


Fig. 4. Enabling notifications for disk usage in Server Admin

Fig. 5. Monitoring disk space with Widgets

Conclusion

The release of Leopard shows us that Apple has been listening and is trying to give us tools that make managing OSX and OSX Server easier. Leopard is new and I am sure we will discover new features as Apple releases them. This is just a short list of items for the system administrator, but there is much more goodness in Leopard that isn't specific to the sys admin.

Items like external home directories can change the way we handle user data along with Time Machine. New versions of directory services, changes to launchd, collaboration software, clustering for VPN services, wide are Bonjour and improved Spotlight are all things that will positively impact our work and the user experience.

As I wrote this article I found more and more little items that I can see as being helpful down the road. It will take some time to find them all and see just how useful they will be, but I am liking what I see. The jury is still out on the simple setup feature of OSX Server, but I don't think too many sys admin's will be using it.

About The Author

Ben has been everything from a Mac user to CTO of one of the leading Macintosh professional services firms. Besides writing an occasional article for MacTech, you can find him presenting at Macworld (including a session called "DNS: Demystified, co-presented with Doug Hanley) or consulting with clients around the world. You can reach him at ben@greisler.org.

by Edward Marczak

Leopard Terminal.app: Create Your Workspace

Have all the comforts of ~

Introduction

Of all the new features that Leopard brings, one might not think that there could be excitement about Terminal.app. One would be wrong. Along with many others, Terminal.app is my primary interface to the system. Yes, I use applications like Word and Mail.app, of course, but the bulk of the time, I'm looking at a green blinking cursor on a semi-transparent black background. I know I don't have to extol the virtues of shell-mastery any more – you've read that enough in this column. However, OS X 10.5, "Leopard," updates Terminal.app in some nice ways. Let's look at the changes, and then ways to customize your workspace so you're making your environment work for you, not against you.

In the Beginning was the Command Line

Perhaps the easiest thing to point out is that Terminal.app has a new icon. It's amazing how jazzed we can all get over an application's *icon*, but that's just part of the Mac way, I suppose.

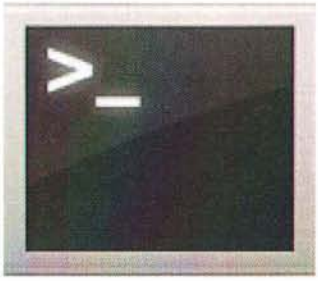


Figure 1 – New Terminal.app icon

It works nicely with the other interface changes in Leopard: a little flatter, but much more stylized.

If you've performed a clean install – which, by the way, I'm a huge advocate of for major releases – and you're not using previous preference files, launch Terminal.app from the Utilities folder and you'll see a default terminal.

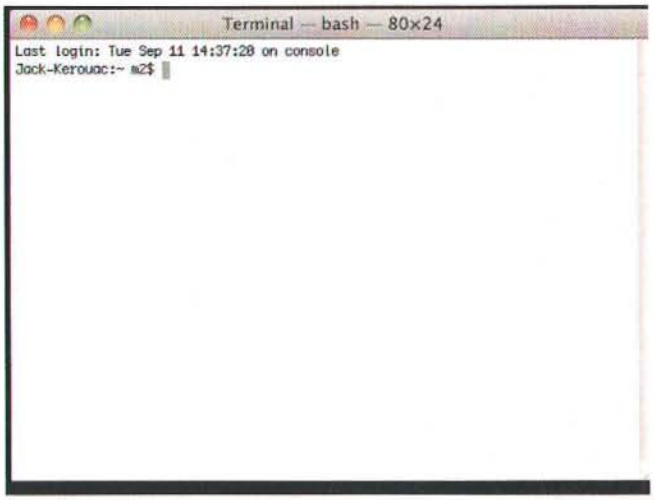


Figure 2 – a console window in its default state.

Well, OK...a little bland, perhaps. Nor will 80x25 cut it in most cases. So, let's start with the easy stuff. Stretch that window out and position it on-screen where it best suits you. I like virtual pagers, like Spaces and VirtueDesktops, so, I highly recommend that Terminal gets its own screen. You may also want multiple terminals open at once, and I'll touch on that a bit more later, but suffice it to say, I basically never have less than 9 *VTs* open at once on my local machine.

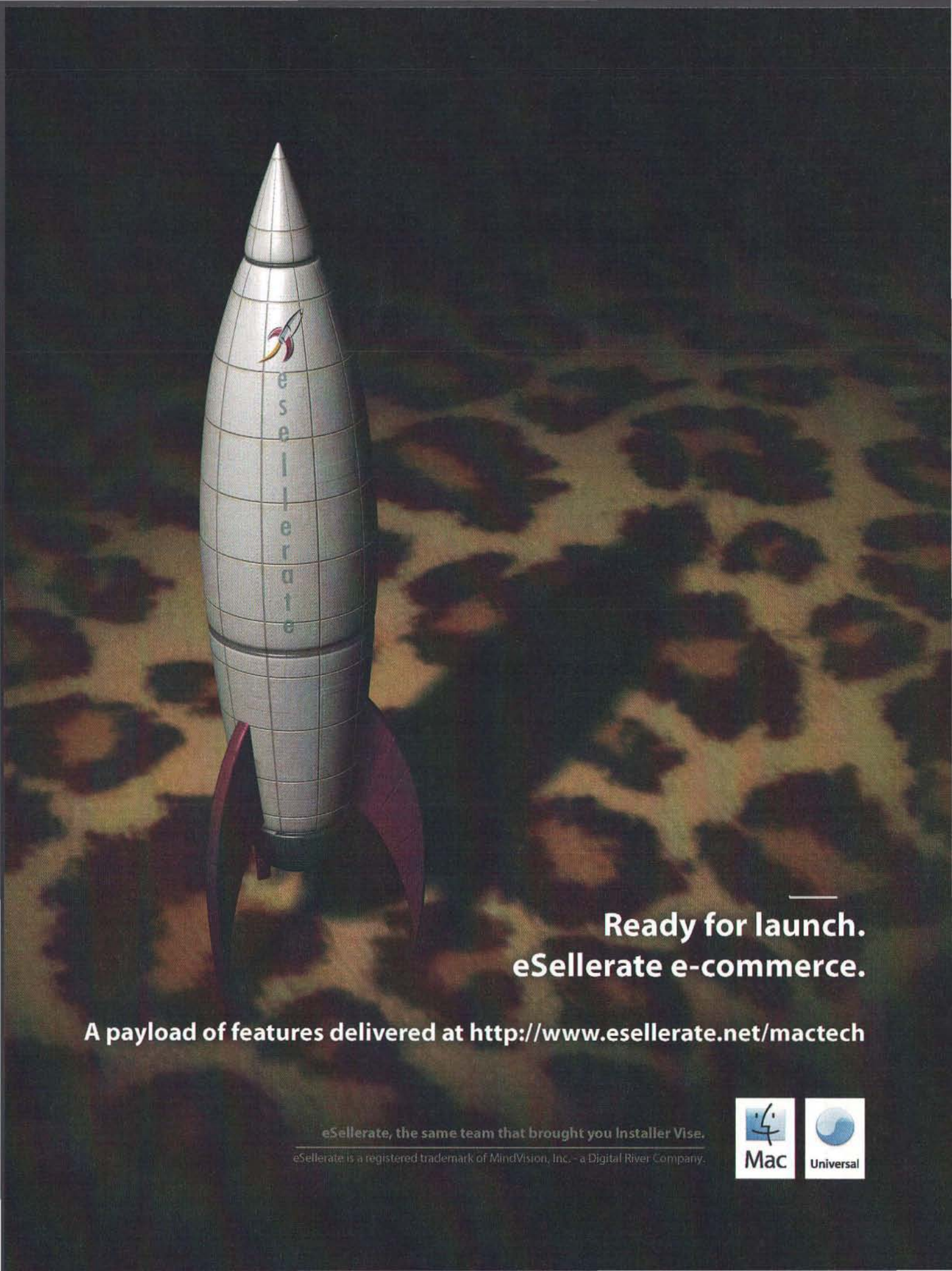
Also easy: if you're familiar with Terminal.app already, you won't be lost. Some of the prefs take a minute to absorb, and some items have simply been renamed (gone is "Connect to Server..." replaced with "New Remote Connection...").

A cry for sanity in our silent night of madness

One Mac In The Shell column that has generated a good deal of feedback had *screen* as its topic. One of Terminal.app's new features happens to be tabs, and most people would think that would do away with `screen`. Not a chance.

If you're in love with tabs, you probably have been using iTerm or Terminator to satiate your needs. Basically, tabbed terminals have existed for OS X, so, this is catch up for Terminal.app. It's a nice addition none-the-less.

Now, I was the first one that thought I'd just ignore tabs in Terminal.app. I already use `screen`, right? Turns out, I find



Ready for launch.
eSellerate e-commerce.

A payload of features delivered at http://www.esellerate.net/mactech

eSellerate, the same team that brought you Installer Vise.

Mac Universal

them both useful. (If you need a refresher on `screen`, check out my original article which now appears on-line at http://www.mactech.com/articles/mactech/Vol.21/21.09/Screen/index.html).

Creating a new tab is as simple as Apple-t, which gives you a new tab in the current window with the default scheme. Or, mouse about to the Shell menu, choose "New Tab", and choose the scheme you'd like. Thankfully, we can change tabs with the same key commands available for tabs in Safari – Apple-Shift-} and Apple-Shift-{. (This goes for tabs in iChat, now, too). So, if your muscle memory is built up like mine, this is a nice addition.

Once any window – tabbed or not – is created, pull up the window inspector using Apple-I, or Shell->Show Inspector to change the properties of an existing window (more detail on Inspector later). See Figure 3 for a look at the Inspector itself, with its settings tab open.

I have `screen` give me 7 VTs at terminal start, and now, one terminal window gives me a two tabs. Why stick with `screen`? First, screen is less resource hungry than Terminal.app's tabs. Allocating a new bash shell under screen just pops up my memory usage by 800k or so. Each Terminal.app tab takes up about 1.5MB. Why the heck am I concerned about this in the age of multi-Gigabyte machines? Well, perhaps, that's just what I do. Plus, despite the 4GB of RAM that my machine has, it still gets strained while running other apps, virtualization and yes, Warcraft (I can quit anytime. Really). So, that memory counts!

Also, when shelling into a remote machine, I typically still have screen at my disposal, and can just fire up a remote screen session. It seems a bit wasteful to open up many local tabs and ssh each one of those to the same remote host. Finally, screen is just wired into my fingers, so, I stick with it for that and its other nice tricks (*and* I get to take my .screenrc file to my new home directory and have everything work as expected – what could be better?).

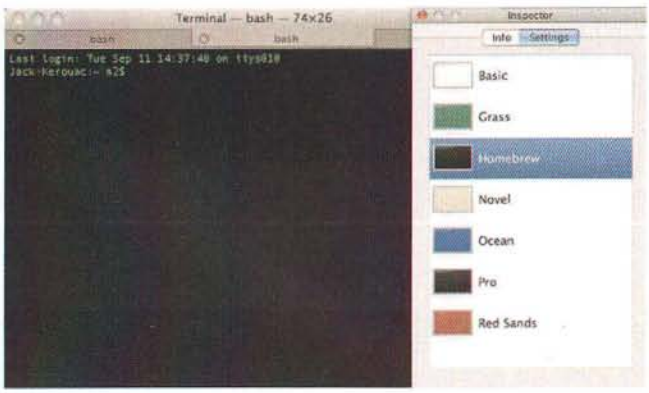
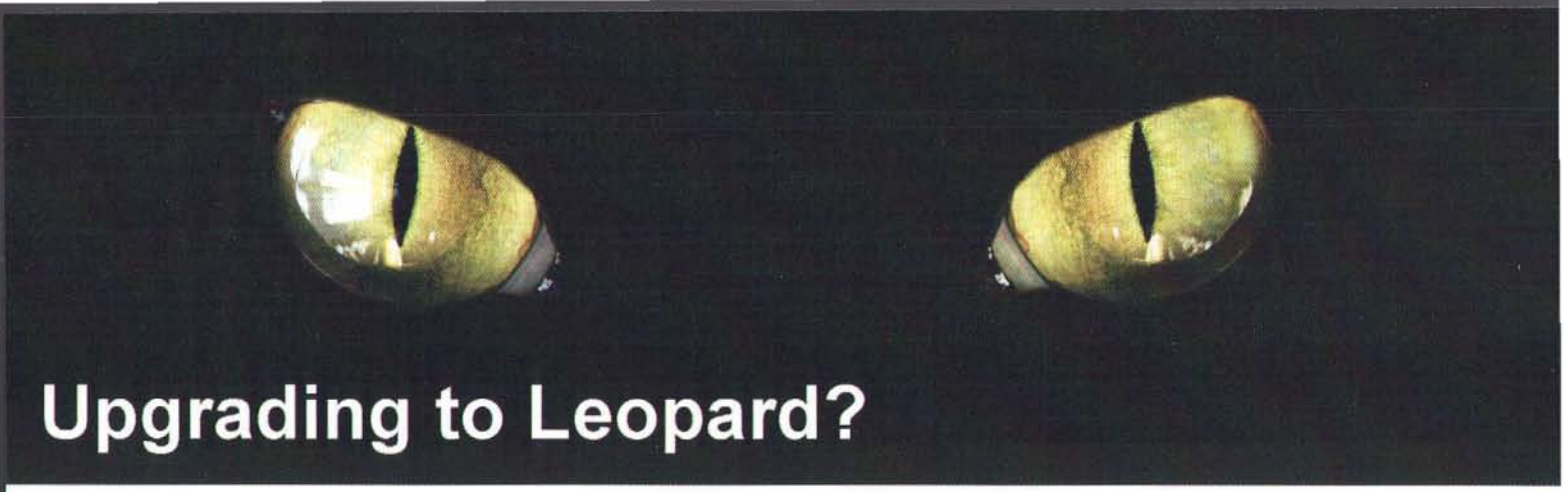


Figure 3 – Terminal tabs and the Inspector window floating over.



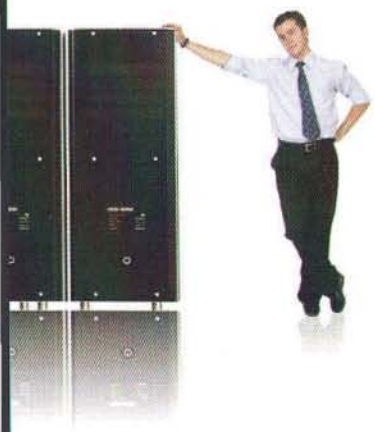
Upgrading to Leopard?

Don't forget to feed the beast.

Upgrade your system memory with Mushkin PerfectMatch to get the most out of OS-X Leopard

It's a jungle out there when it comes to operating system upgrades and the ever increasing minimum system requirements for optimal performance. Now, with Mushkin PerfectMatch memory, you won't have to play guessing games and choose between the increased functionality of Leopard and your current version of OS-X.

Because every PerfectMatch Apple memory module is built from components carefully selected to provide true compatibility and match or exceed the performance of the original Apple memory installed at the factory. Which means you can tame the beast and confidently and easily upgrade the performance of your system to run Leopard.



mushkin **PerfectMatch**

Memory to match your system perfectly.

- ***Guaranteed, 100% Compatibility***
  Every PerfectMatch module is guaranteed to match or exceed the performance of the original memory installed at the factory.
- ***Save hundreds of dollars over the price of a new Mac***
  Mushkin PerfectMatch memory upgrades provide an easy and inexpensive way to optimize overall performance.
- ***Taking care of your company's road warriors?***
  Now you can cater to your laptop users' voracious appetite for RAM and enhance their mobile computing lifestyle.
- ***Your creative genius just met its match.***
  Upgrading with PerfectMatch memory increases performance in memory-hogging multimedia applications.

Now That Cat Can Run.

v i s i t u s o n l i n e a t w w w . m u s h k i n . c o m

North America
333 W Colfax Ave Ste 400
Denver, Colorado 80204
United States of America
tel: (800) 569-1868
fax: (303) 534-0827

Europe, Middle East & Africa
Raiffeisenstr.9
76872 Minfeld
Germany
tel: +49 7275-9888-13
fax: +49 7275-9888-69

Asia & Pacific
B-13-9, Megan Avenue II, No12
Jalan Yap Kwan Seng
50450 Kuala Lumpur
Malaysia
tel: +60-12-3157008

<m Get More.

the color of television, tuned to a dead channel

Everyone has their own idea of what a perfect terminal screen should look like. Me? I like the 'classic' look of green (or even amber) on black. That's just the way it was. But I also have a 'classic' Commodore 64 looking terminal (blue on blue). While this may seem trivial at first, you're going to be staring at these screens *a lot*. Also, you're going to need to pick out text from the rest of the clutter on screen. So, find a color scheme that you like and a font that suits you. This has been made easier with the current incarnation of Terminal.app.

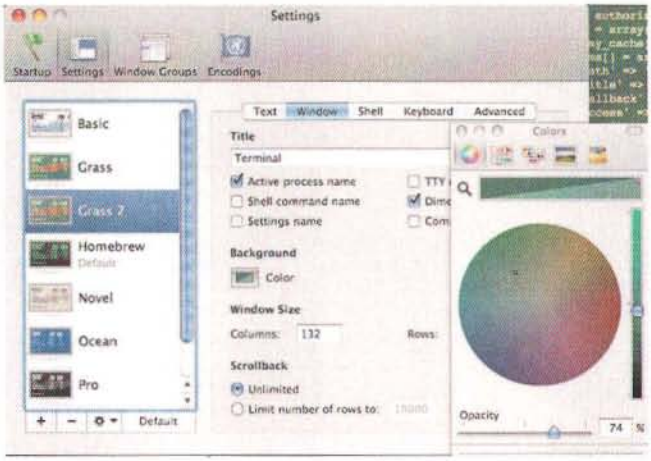


Figure 4 – Preferences and Color Picker

While slightly confusing at first, the new preferences really are simplified. Gone are the separate "Preferences" and "Window Settings" menu choices. Now, all 'preferences' are consolidated. Colors, fonts, window sizes and more can be found under the "Settings" group. You can easily define groups of styles that you like and save them for future use, and even export the settings for your or other's use.

One exceptionally nice touch is that if you have a saved terminal definition from, oh, you know, the *Tiger* days, just double-click it. Not only will you be looking at your familiar terminal, but Terminal.app 2 will import the setting, making it easily called up in the future. To this I say, thankyouverymuch!

Additionally, if you like to have multiple terminal windows open at the same time, the new Window Groups feature is a blessing. All you have to do is get your terminals arranged as you like them: color, fonts, size and *position*. Then, that can be saved as a Window Group. Simply use the "Save as Window Group..." menu item from the Window menu, or, add it with the "+" button in the Window Group section in Preferences.

SQL Connect in FileMaker® 9

Here's Why Servoy is a Better Tool:

If you're using FileMaker Pro 9's new ability to connect to external SQL databases - or you're considering an upgrade - you owe it to yourself to have a look at Servoy.

Why? Because FileMaker is not a "true" front end for SQL databases. In fact, they're the first to say so in their own white paper.* Servoy was built from the ground up to give you a true front end into your SQL database(s).

Servoy doesn't force you to use proprietary ODBC drivers to connect to a limited number of SQL sources - you can connect to any SQL database with any JDBC driver you choose.

Because Servoy isn't a database - it doesn't have to rely on workarounds like "shadow tables" and "shadow fields" to interact with the SQL data - Servoy will just directly connect to the datasource.

Servoy supports server-side sorting and user-defined SQL queries so you get maximum performance and maximum flexibility when dealing with the back end databases.

Servoy will automatically take care of "broadcasting" the changes that users make to other users, to help ensure the consistency of the data. FileMaker 9 doesn't.

Servoy allows you to have more than 100 simultaneous users and will scale to whatever level you need based on hardware - it doesn't have the 100 connection limitation of FileMaker Server 9 Advanced.

Servoy specializes in efficient, fast SQL code when talking to your SQL sources. The free Servoy Application Server has built-in connection pooling, performance history, and supports inner and outer SQL joins.

In a nutshell - Servoy was designed to work with SQL sources in the most efficient, flexible and seamless way. FileMaker 9 has added to the types of external files that you can connect to - but you can't touch, view, optimize or customize the SQL they generate.

Which tool would you rather use?

SERVOY®

Download a FREE trial at: **www.servoy.com/mactech**

*Based on the FileMaker, Inc. white paper entitled "Introduction to External SQL Sources" available from the www.filemaker.com website as of July 16, 2007.

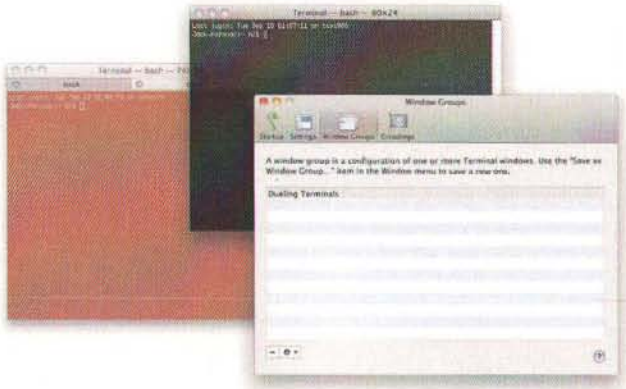


Figure 5 – Saved Window Groups

Finally, you can choose a Window Group to appear when you launch Terminal.app. What a fantastic feature!

As far as all of the other Terminal.app accoutrements that you've gotten accustomed to, they're all still there. For anyone in love with Transparency, take note where the opacity slider has moved to. Interestingly, this lets you set opacity on *any* color element, including text and your highlight color. In other words, you can either end up with something very cool looking, or something that impossible to read!

It is by will alone I set my mind in motion

What else has changed? There is one topic I haven't covered before in this column that I've had some people ask me about recently. Back in the day, when we would be talking about *physical terminals*, the host system had to know the terminal's capabilities. The terminal that you were using possibly had a real, hardware limit on how many columns it could display, if it supported color or blinking text, graphic support, etc. Ultimately, each terminal's capabilities got catalogued and placed into the *termcap* database. Fast forward to a contemporary setup, and termcap has become *terminfo*. Same idea, different name. Figure 6 shows where in Preferences->Settings we declare which terminal type we're emulating (what we're pretending to be).

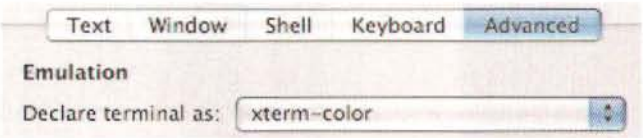


Figure 6 – Declaring our terminal type

The killer app for the Mac is better than ever!

Parallels® Desktop 3.0 for Mac is here.

With Parallels® Desktop 3.0 for Mac, you can:

- Automatically open Windows files with Mac software and Mac files with Windows software
- 'Bulletproof' your virtual machine from Windows system crashes and malware
- Run selected Windows-only 3D games and applications with 3D graphics support
- Browse through Windows files and folders without launching Windows

SWSOFT PARALLELS

Parallels® Desktop 3.0 for Mac provides even more ground-breaking features and capabilities than previous versions. Already the top-ranked software on PC World's "Best 100 Products of 2007" list, Parallels® Desktop continues to drive innovation in desktop virtualization.

Want better Windows and Mac integration? Want to run the hottest PC games and graphics software? Worried about security and system mishaps? Parallels® Desktop 3.0 for Mac includes 50+ new features and enhancements, including a number of integration features and virtual machine utilities unavailable anywhere else.

To discover why Parallels® Desktop is the leading solution for running Windows on a Mac, visit www.parallels.com or call us at 1 (425) 282-6405.

PC WORLD 100 BEST PRODUCTS OF 2007 — Macworld BEST OF SHOW 2007

Download a trial today at www.parallels.com/download

Website – www.parallels.com Email – sales@parallels.com Phone – 1 (425) 282-6405

Interestingly, if we're running `screen` – I use it as my shell – we get a terminal type of `screen`. There also happens to be a terminfo entry for screen, declaring the capabilities your terminal will have.

Ever notice how after looking at a man page, or say, after using vi, you get the contents of your terminal back. Just like you had never run man or vi? Other programs will do this for you, too. Some physical terminals had an alternate screen buffer that apps are directed to, and when they quit, the main buffer is swapped back in. The xterm-color terminal in OS X mimics that. So, you're poking around the filesystem using cd and ls nad find a text file that you need to edit. Your terminal window is littered with browsing history and the prompt is at the bottom of the window. You fire up vi, edit the file, save and quit. Lo and behold, you're back where you were before editing. The scrollback buffer shows no evidence of the vi session.

`screen`, coincidentally, has its *own* scrollback buffer that overrides terminal's. So, as items scroll past the top of the window, they seem gone. Using screen's scrollback (ctrl-[by default), you can go back through terminal output. Personally, I like my two-finger scroll, despite the mish-mash you get in the process. Apparently, many others do, to, as the first question I get from people that I turn-on to `screen` is, "why can't I use the scroll bar anymore?"

This is an easy one with a nice fix – "nice" meaning that you can alter the behavior for you, not touch the system defaults, and furthermore carry the change around with you in your home directory. `screen` lets you override the terminfo entry for a given terminal. Simply place the following line in your .screenrc file:

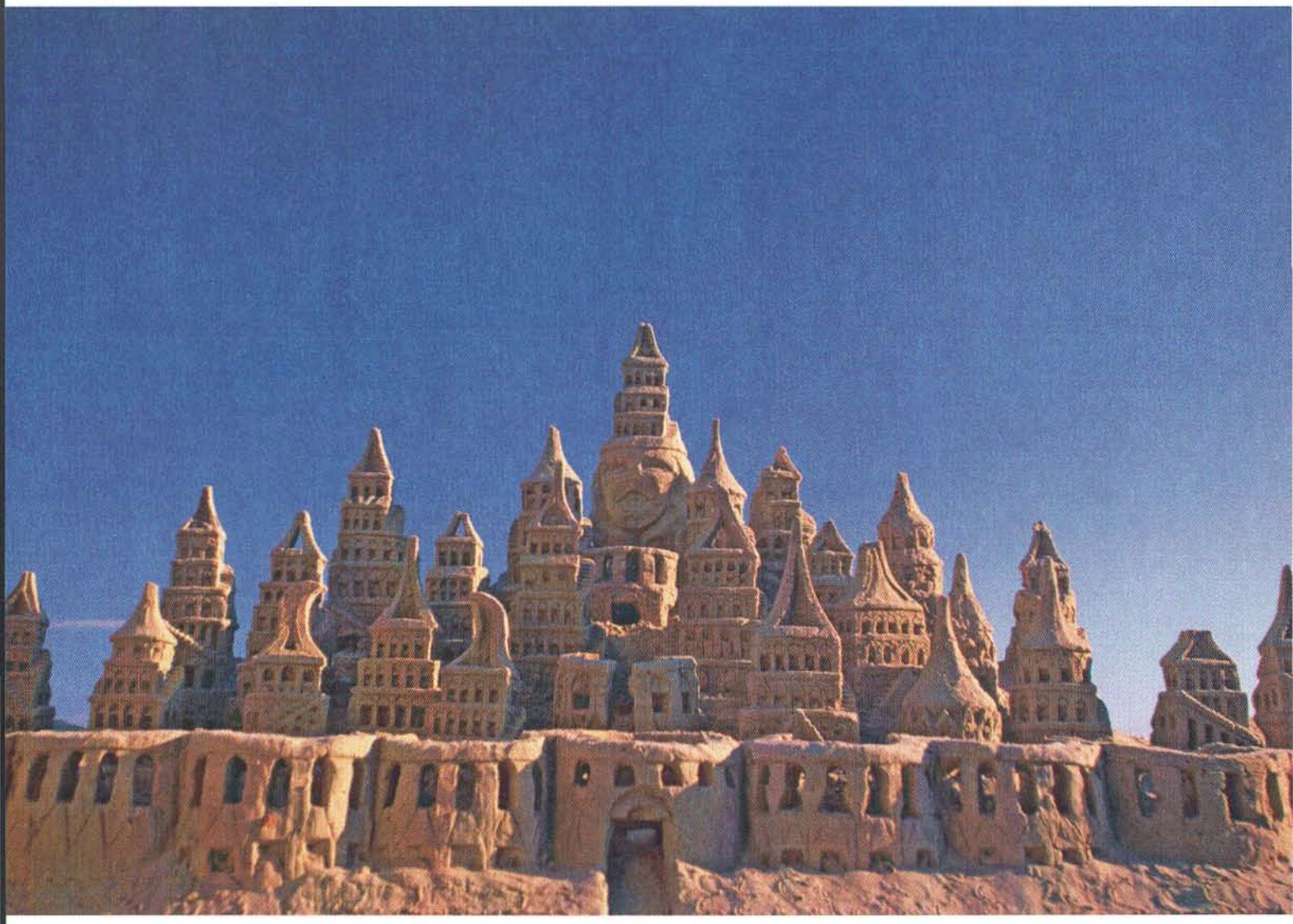
```
termcapinfo xterm* ti@:te@
```

Bail out of `screen` if it's already running and fire it up again. Use top. Go ahead. Now quit top. All of that output stays plastered all over your terminal. Beautiful, beautiful output! That line simply tells any xterm running under screen that it no longer has the capability of an alternate buffer. It basically redefines the escape codes. If more of this type of skullduggery intrigues you, start with the term(5) man page.

There's a frood who really knows where his towel is

Terminal 2 really brings some nice touches and conveniences to those of us who use a command shell as their primary interface to the system. Even if you don't use it as often as others, getting deep into the system as a developer or sysadmin means breaking out Terminal.app on more than a

imagine. amaze. inspire.



you think it. you create it.

www.realbasic2007.com
Try REALbasic 2007 Release 4 risk-free!
REALbasic includes a 90-day
money back guarantee.

REALbasic is the powerful, easy-to-use tool for creating your own software. At REAL Software, we call it a problem solver. You've probably said, "Wouldn't it be great if there was a little application that ..." REALbasic fills that blank.

REALbasic compiles native applications for Mac OS X, Windows and Linux from one set of source code. Each version of your software looks and works just as it should in each environment. You can even create a Universal Binary with one mouse-click.

REALbasic®

passing occasion. By this point, you should have a productive and attractive (to you) workspace that works with you.

Media of the month: I should probably be recommending something *Leopardish*. I can't just yet. But, I was in a book store (again) just the other day and realized how long it had been since I sat down and read fiction. Made up characters, made up places and made up tales. You need to be in touch with reality, but exercising creativity also ranks pretty high. If you've never ready anything by William Gibson, that's my recommendation. Find one that you like. If you've never read anything by Neil Gaiman, please, hurry to your local bookshop and pick up "American Gods" or "Neverwhere." And if you've never put eyes to paper that contains "The Hitchhiker's Guide To the Galaxy," then, run! Go now!

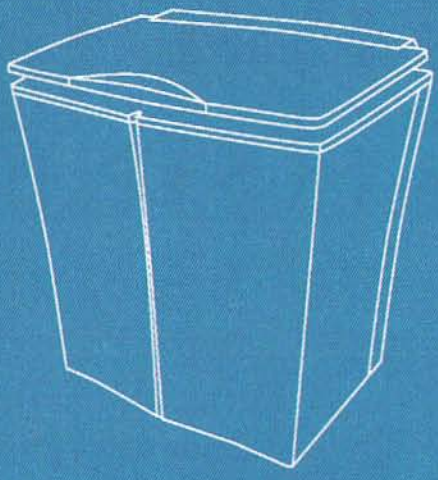
We're also getting close to Macworld. So close that if you haven't made your plans yet, you may be too late. Fire up your web browser, purchase tickets, your travel and hotel for this worthwhile event. Hope to see you there!

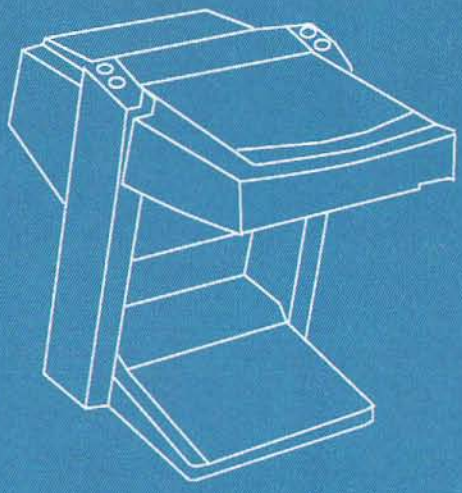
'Mt

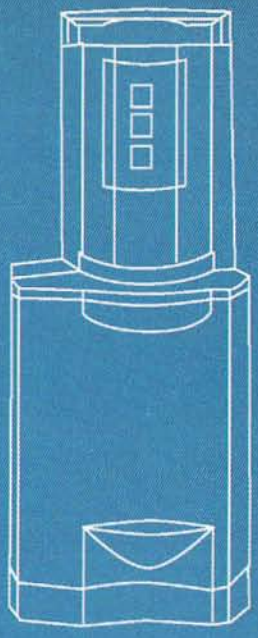
About The Author

Ed Marczak is just this guy, you know? Find out why at http://www.radiotope.com

SilverFast® Ai Studio



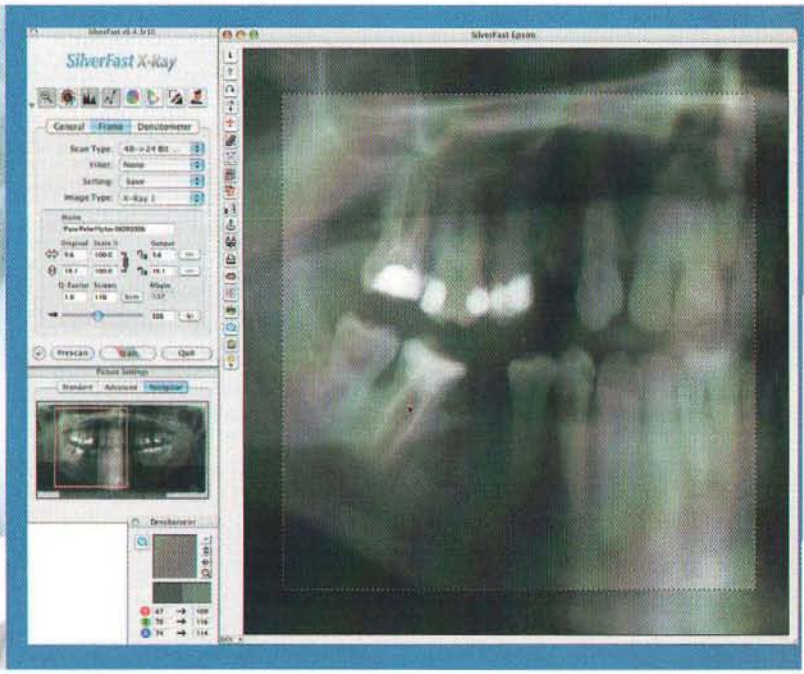




SilverFast Ai Studio now available for Heidelberg HighEnd scanners
(Mac OS X fully supported on any Mac - including IntelMacs)
Topaz, Tango, PrimeScan, NexScan, Chromagraph

SilverFast® X-Ray

Software for medical radiographs
available for most Epson scanners



For more information, please contact boris at: boris.bischof@silverfast.com

LaserSoft Imaging®

www.SilverFast.com
Demo Versions available for all versions!

newegg.com®

ONCE YOU KNOW, YOU NEWEGG.®

[Enter Promo Code MCT11207 for a FREE T-SHIRT with these items]

GRIFFIN

$129.99

Griffin AMPLIFI 2.1 Sound System for iPod

- Compact acoustically-tuned wooden enclosure
- 3.5mm stereo input jack supports all players
- 5" woofer and bass reflex port
- Slim 6-function IR remote
- Charges iPod during use
- Item# 55-999-267

• 99% of orders ship within 1 business day • Over 750,000 product reviews from customers • "A" rating on ALL merchant ranking sites • Over 50,000 customer testimonials

LACIE

$188.99

Lacie d2 Quadra 500GB eSATA External Hard Drive

- Quadruple interface for professional use
- Advanced cooling system
- Fast interface transfer rates up to 1.5Gbits/s
- 7200 RPM
- 16MB Cache
- Daisy-chainable via FireWire
- Item# 22-154-178

LACIE

$314.99

Lacie Big Disk Extreme+ 1TB External Hard Drive

- Triple interface for universal connectivity
- Built-in RAID 0 for extreme FireWire 800 speed
- Compact design stack, rackmount, stand upright
- 7200 RPM
- 32MB Cache
- Lacie Shortcut Button
- Item# 22-154-185

Klipsch

$99.99

Klipsch iGroove HG iPod Speaker

- Dual 2.5 inch woofers
- Dual 1 inch Horn-loaded tweeters
- Charges iPod while docked
- Integrated handle
- IR handheld remote control
- Works with most MP3 player brands
- Item# 55-998-130

iHome

$99.99

iHome Dual Alarm Clock Radio with dock for iPod

- Wake to iPod, AM/FM Radio, or Buzzer
- Sleep to your iPod or AM/FM radio
- Compatible with iPod's with dock connecto
- Plays and charges docking iPod models
- EQ controls treble, bass, and balance
- Listen to other AUX audio sources
- Item# 55-984-017

NO PAYMENTS FOR 6 MONTHS with your *Newegg.com* Preferred Account

See website for more information. Promotion ends 01/31/08

WINNER 3 YEARS IN A ROW!
BEST PLACE TO BUY
- ONLINE - PERIPHERALS
- SOFTWARE - CORE COMPONENTS

COMPUTER SHOPPER SHOPPERS' CHOICE 2006

Balancing Great Products with Superb Service

Make Newegg.com your destination for savings.

MINI COOPER

Enter the Canon "Mini" Holiday Sweepstakes!

(Sweepstakes ends 11:59pm PST on 12/31/07. Check website for more info)

CREATIVE

(iPod not included)

$119.99

Creative TravelSound i Portable speakers for iPod

- One-piece portable speaker system
- Two NeoTitanium MicroDrivers per channel
- Auxiliary input works with any MP3 Player
- Plays and charges docking iPod models
- Wireless remote control included
- Protective carrying case
- Item# 55-996-033

BELKIN

$39.99

Belkin TuneCast Auto for iPod

- Transmits on FM channels 88.1MHz - 107.9MHz
- Powers and charges through your car's cigarette lighter
- Two Programmable memory presets
- Includes surface mount clip for positioning
- Compatible with select cases
- Item# 55-999-450

DLO

(iPod not included)

$59.99

DLO Transpod In-Car Cradle/FM Transmitter

- Play, charge and cradle your iPod
- Transmits iPod music to your car's FM stereo
- Charges iPod while securely cradled
- 4 assignable Presets
- Aux-input lets you connect any MP3 Player
- No install necessary
- Item# 55-998-073

macally

$9.99

Macally Cassette Tape Car Adapter for iPod

- Integrate your iPod into any car cassette player
- Listen to your MP3 music on the road
- High fidelity and reliability
- 3.5mm connector w/ 4' cord
- Compatible with all iPod models
- Supports players w/ 3.5mm headphone jack
- Item# 55-998-405

The Road to Code

by Dave Dribin

Leopard Detour

What's new for developers in Leopard

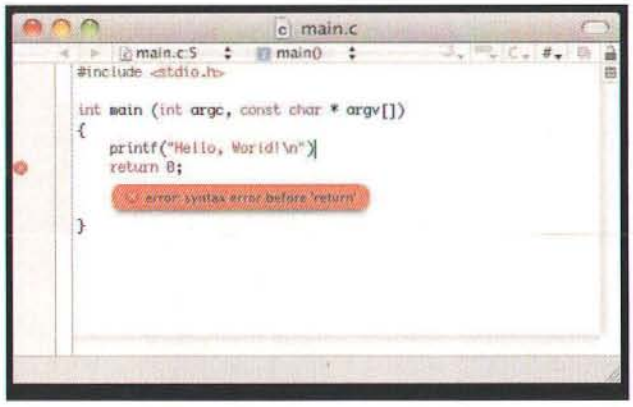


Figure 1: Xcode error bubbles

Short Detour

In this month's *The Road to Code*, we're going to be taking a short detour from the usual programming road. Instead of talking about Objective-C and object-oriented programming, I'm going to talk about Leopard. You may have heard about Apple's next generation operating system, Mac OS X 10.5, a.k.a Leopard. While much of the marketing buzz revolves around the new features available to the end-users, such as Time Machine, Spaces, and Cover Flow in The Finder, many improvements are in store for the developer, too. Since I still have not used Leopard extensively for development and there are far more changes than I can discuss in a single article, I'll be picking out some of the highlights that I've already encountered or am looking forward to using. While this article will be most useful for existing developers, you'll still learn a lot about developing for Mac OS X, even if you're not an old hat.

Development Tools

There are many improvements in the existing development tools, but there are also a couple of interesting new ones.

Xcode

Leopard comes with a new version of Xcode, Apple's integrated development environment (IDE), now up to Version 3.0. The first change you'll notice is that the main text editor has received an overhaul. The most immediate difference you'll encounter is that errors and warnings are now displayed inline as bubbles of text, as in Figure 1. While this is better than constantly keeping the build window open, or hovering over errors in the gutter to see the full message, it can also be a bit distracting. The bubbles shift lines of text down to fit themselves on screen, and I find this makes it a bit hard to see context around the error in question. For now, I'm keeping this option enabled, as it may grow on me over time.

The debugger has been better integrated, so that debugging doesn't require as much mental context switching. One of the best features is that the debugger can be invoked any time you run the application. There are no longer separate Build and Run and Build and Debug options. This is wonderful because often you are testing a program, only to realize that you'd like to set a breakpoint when something goes wrong. In the previous versions, you'd have to re-start the program under the debugger. Now, you can just set the breakpoint, and the debugger will automatically attach to your running application. In addition, there's no longer a separate debugger window, which helps keep you focused on the problem at hand.

Xcode 3.0 also provides the ability to refactor Objective-C code. Refactoring allows you to make your code more readable or simplifies its structure without changing the behavior. Other IDEs have had refactoring for a few years now, and it is a feature I am really excited about. Some examples of refactoring provided by Xcode are renaming variable names (including instance variables), renaming classes, extracting a block of code into a method, and the ability to move methods up and down in the inheritance hierarchy. While some of these can be done with search and replace, refactoring, in addition to being easier, is usually more accurate since it is not purely text based.

The long laundry list of changes does not stop there. Here's a taste of more new features:

Code folding and code focusing using a focus ribbon.

Code completion has been simplified with inline code completion.

Project snapshots allow you to experiment freely and revert code changes, if necessary. While I don't see this replacing a full-blown source code management (SCM) system, it is still nice to have.

A better Build Settings editor, including the ability to set architecture-specific settings.

The Research Assistant provides easy access to reference documentation, without taking your focus away from the code.

SnapzProX

Now Intel Native!

If a picture is worth a thousand words, imagine how priceless a movie would be...

Snapz Pro X allows you to effortlessly record anything on your screen, saving it as a QuickTime® movie that can be emailed, put up on the web, or distributed however you want.

Why take a static screenshot when Snapz Pro X makes creating a movie just as easy?
In addition to a video capture engine that is 20 times faster than anything else on the market, Snapz Pro X has so many other new features, you'll quickly wonder how you ever lived without it!

Download your free trial now:
http://www.AmbrosiaSW.com/MacTech

AMBROSIA® SOFTWARE INC

2004 winner macworld eddys

Macworld

Mac

Universal

Adds the ability to have versioned Core Data models. This helps data migration as your model changes over time.

GUI access to SCM repositories.

For details, see the "Xcode 3.0 Feature Overview" document.

Interface Builder

Interface Builder also reaches Version 3.0 with significant updates. Interface Builder (IB) has a long history, going all the way back to NeXTSTEP (the `.nib` extension is short for NeXT Interface Builder), and was ripe for an update. The user interface has been completely updated for 3.0. Separate palettes for GUI controls have been replaced with the concept of a Library. All controls are now shown in a single list, as in Figure 2. While the Library still groups similar controls, you can also search for controls by name. The list dynamically updates to show only the matching controls.

Figure 2: Interface Builder Library window

IB also adds support for some new controls, but the one I am currently most excited about is support for `NSToolbar`. Toolbars have previously needed to be setup programmatically, which can be a bit of a chore. Now, you can add a toolbar to a window, add items to a toolbar, and hookup their actions all in IB.

The Inspector Window has been improved to support multiple selection. This a is big time saver, since you can now set properties of multiple controls all at once, rather than selecting each one individually. The best part, I think, is that you can select any set of controls, and only the common properties are shown.

My favorite update, though, is the ability for IB to automatically synchronize outlets and actions to header files. Gone are the days where you drag and drop changed header files from Xcode to IB to synchronize them. This single update alone will probably save you hours of frustration.

For a full list of changes, see "Interface Builder 3.0 Release Notes," but here's a quick list of other exciting changes:

Support for Core Animation.

Support for Tree Controller and Dictionary Controller.

Real-time animation to visualize springs and struts.

Interface Builder Plug-in model to replace `IBPalette`.

DTrace and Instruments

DTrace is a fantastic performance diagnostic and system tracing tool, originally developed by Sun Microsystems for their Solaris operating system. Sun released it as Open Source, and Apple has ported it to Mac OS X for Leopard. DTrace is a command line application that can be used not only to monitor system performance, but also as a debugging tool. For example, you can print out arguments of functions when they are called with very little overhead, and without recompiling. DTrace is extremely flexible and even has it's own scripting language called D, so you can use to write custom tracing scripts. And DTrace isn't just for user applications. You can use DTrace on the kernel and your own kernel extensions, too.

But the goodness doesn't stop there. Instruments is a new GUI performance monitoring tool, reminiscent of Garage Band, but for developers. You can add different "instruments" such as memory usage, CPU load, and object allocation, and you can see these measurements change over time. Under the hood, Instruments uses DTrace and it's essentially just a pretty face on top of `libdtrace`, the programmatic API for DTrace. This also means you can customize Instruments by writing your own instruments using D scripts.

I have not used DTrace or Instruments extensively but I am willing to bet that they will find a permanent home in my tool belt, along side Apple's already fantastic list of performance tools, such as Shark.

The Missing Sync™

Reliable Handheld Synchronization for Mac

Thanks to **The Missing Sync**, Mac users are no longer second-class citizens when it comes to keeping their mobile devices in sync. Whether it's a **Windows Mobile 6** device, a **BlackBerry** or even a **Palm** Treo, there's a Missing Sync product made to connect and synchronize that device with Mac OS X.

Syncs Contacts, Calendars, Tasks

- Supports Address Book, iCal, Microsoft Entourage 2004
- Outstanding field support, even syncs Address Book contact photos
- Supports calendar event reminders and detached events
- Sync Services-savvy for syncing with third-party apps

Complete Notes Syncronization

- Includes Mark/Space Notebook for Mac to create, edit, categorize, sort and search through notes
- Also supports Microsoft Entourage 2004 and Bare Bones Yojimbo
- Full support for Sync Services and .Mac syncing between Macs

iPhoto and iTunes Integration

- Imports photos and videos from mobile devices into iPhoto albums or folders in the Finder
- Resizes and downloads selected iPhoto albums to the device for handheld viewing and slideshows
- Downloads DRM-free music and podcasts for mobile playback

The Missing Sync for BlackBerry, Palm OS or Windows Mobile is available in single-user licenses for $39.95 or in multi-user packs for any size organization, The Missing Sync family of products provide a Mac-centric synchronization solution second to none.

Visit www.markspace.com today to see how easy it is to sync the latest smartphones and mobile devices with the Mac.

mark/space

Objective C 2.0

The native language for Mac OS X applications, Objective-C, also gets a new major revision, giving us Objective-C 2.0. In Objective-C 2.0, we'll see fast enumeration, garbage collection, simplified property accessors, and enhancements to protocols.

Fast enumeration

Fast enumeration is combination of syntactic sugar and a speed increase. The old school way of looping through all objects of an `NSArray` is using an `NSEnumerator`, as follows:

```
NSArray * colors = [NSArray arrayWithObjects:
    @"red", @"green", @"blue", nil];
NSEnumerator * e = [colors objectEnumerator];
NSString * color;
while (color = [e nextObject])
{
    NSLog(@"Color: %@", color);
}
```

The new way to do this in Objective-C 2.0 is to use a special `for` loop syntax:

```
for (NSString * color in colors)
{
    NSLog(@"Color: %@", color);
}
```

Not only is this easier to read, but it is apparently much more efficient than good ol' `NSEnumerator`. You can even add fast enumeration to your custom classes if you implement the `NSFastEnumeration` protocol.

Garbage Collection

One of the trickiest aspects of programming in Objective-C is dealing with dynamic memory management. Although reference counting is an improvement over `malloc/free` in C and `new/delete` in C++, it is still cumbersome and error prone. With Objective-C 2.0 comes garbage collection (GC) and the promise of making dynamic memory management painless. At a high level, this means you no longer have to send `retain` and `release` messages to objects to update their reference count. It also means you do not have to implement `dealloc` to free your instance variables. The garbage collector will find unused objects, affectionately known as garbage, and free them for you. It also simplifies your accessor methods. For example, instead of implementing `setWidget:` as follows:

```
- (void) setWidget: (Widget *) widget
{
    if (_widget != widget)
    {
        [_widget release];
        _widget = [widget retain];
    }
}
```

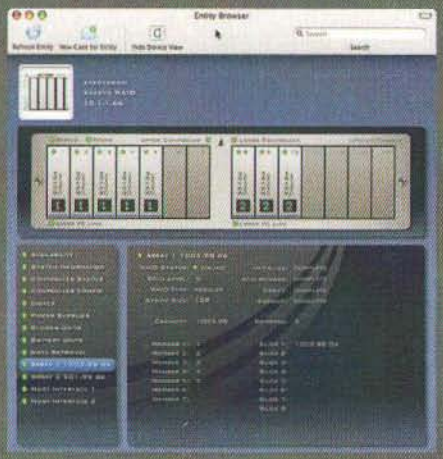
You need to simply do an assignment:

```
- (void) setWidget: (Widget *) widget
{
    _widget = widget;
}
```

The garbage collector also handles retain cycles properly, so you no longer have to deal with those specially.

Network, Server and Appliance Monitoring.

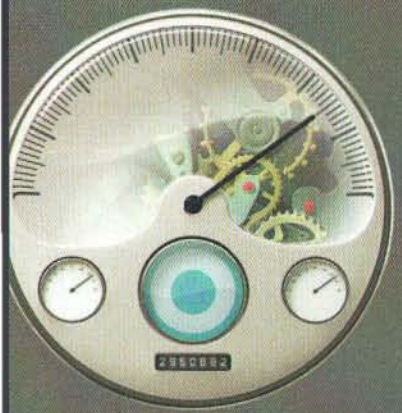
For Mac OS X



Xserve RAID

Xserve (Intel & G5)

Airport



Lithium Network Monitoring Platform

Lithium can now monitor your Xserve, Xserve RAID, Qlogic switches, Airports, Mac OS X Server... and everything else in your network.

www.lithiumcorp.com

lithium Corp.

This is common in parent/child relationships and using delegates.

There is quite a bit more detail to garbage collection, especially if you are converting an existing code base, but this gives you a taste of what is available. It's also worth noting that Xcode 3.0 is itself a garbage collected application. I think this is a good sign for Apple to eat it's own dog food and it gives me a warm fuzzy that GC really is the way forward for new, Leopard-only applications. I think it will also quell the gut instinct of many who believe that GC impacts performance too much. As I usually say to people who worry about performance: profile it before blaming something. I'm sure we'll see more on this topic as Leopard gets thrown through the ringer in the months to come. Overall, I'm really excited to see get my hands dirty with GC, and I think it will make the learning curve for newcomers to Objective-C a little less steep. Oh, and in case you're wondering how all of your existing applications will work, garbage collection is disabled by default, so no worries.

Properties

Continuing the trend of making the programmer's life simpler is the introduction of properties. The concept of properties is not new to Objective-C and is the backbone of Key Value Coding (KVC), Key Value Observing (KVO), and Cocoa Bindings. After writing even a small amount of Objective-C code, you will find yourself writing a lot of boilerplate code for KVC-compliant getter and setter methods, affectionately know as accessors. As I showed above, implementing accessors has been simplified with garbage collection. Nevertheless, it's still a lot of brainless code to write. You have to implement two methods for each property, and if you can't use garbage collection in your application, it's even more brainless code. Objective-C 2.0 now has special syntax to work with properties.

New keywords are provided to simplify the declaration and implementation of properties. First, the declaration:

```
@interface Person : NSObject
{
    NSString * _name;
}

@property(copy, readwrite) NSString * name;

@end
```

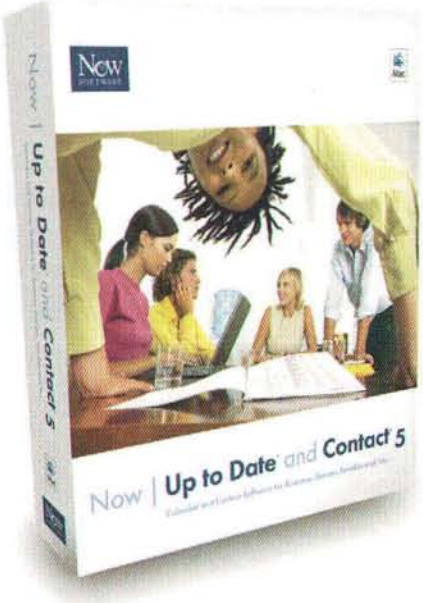
The `@property` line is similar to declaring `name` and `setName:` methods. It states that there is a property named `name` of type `NSString *`. The attributes given in the parenthesis customize the behavior of the property. In this case, `copy` means perform a copy in the setter, and `readwrite` means declare a getter *and* setter. A property can be `readonly` if there should be no setter method. There are other property declarations, too, so please see "The Objective-C 2.0 Programming Language" document for details.

On the implementation side, there is a new keyword to generate the method implementations for you:

Now **scheduling** and **contact management** for your entire organization.

Now | **Up to Date**® and **Contact**® **5**

Calendar and Contact Software for Business, Groups, Families and You.



Is this project on schedule? When are you available to meet about the systems upgrade? Where are all the field techs today? When was the last time anyone talked to our biggest customer?

Virtually all groups live (or die) by their abilities to meet deadlines and keep track of their customers, prospects, and vendors. Few small companies or even departments of big companies have the tools they need.

Now Up-to-Date & Contact might just be the calendar and contact software for you. It's time-tested and used by more Mac-based companies than any other solution. And it's cross-platform–available for your PC users, too. It's easy to install and manage and simple for your employees to understand and use.

Using Now Up-to-Date & Contact you can schedule meetings for multiple users, view multiple, simultaneous calendars, and reserve rooms and resources. You can share contact information about your customers, prospects and vendors. And using our free server software you can set it up in minutes and share with users in the office or from anywhere with an internet connection.

Call us now at 866-527-0556 or email us at mactech@nowsoftware.com and we'll send you our free evaluation kit, including the book that will make it all easy, "Take Control of Now Up-to-Date & Contact" from Take Control books!

NOW SOFTWARE

Phone: 866-527-0556

Web: www.nowsoftware.com

```
@implementation Person

@synthesize name = _name;

@end
```

The `@synthesize` keyword implements the methods for property named `name` using the instance variable `_name`. Again, there are different ways to customize the property implementation so see the official manual for details.

The end result of using the property keywords is a lot less boilerplate code. But wait, there's more! Objective-C also provides a new dot syntax to simplify property access. Thus, instead of invoking the accessors methods as follows:

```
Person * person = ...;
NSLog(@"Name: %@", [person name]);
[person setName: @"Steve Jobs"];
```

You can use the dot syntax:

```
Person * person = ...;
NSLog(@"Name: %@", person.name);
person.name = @"Steve Jobs";
```

Protocols

The last feature of Objective-C 2.0 I'm going to introduce is an enhancement to formal protocols. Protocols define a set a methods that implementing classes must implement. Now methods can be defined as optional using the new `@optional` keyword:

```
@protocol MyProtocol

- (void) requiredMethod;

@optional
- (void) anOptionalMethod;

@end
```

Since optional methods are often seen in informal protocols, such as delegates, I think this feature will get used a lot. The benefit to using a protocol with optional methods over an informal protocol as that it makes your code more explicit and readable. It also improves the runtime metadata about an implementing class. For example, at runtime you can check to see if an object conforms to a formal protocol using `conformsToProtocol:`, but not an informal one.

Objective-C Runtime

The features listed above are for the Objective-C language syntax, but the runtime underpinnings also get an update. Mostly, this is transparent, but if you do any low-level hacking, you'll want to read up on the new runtime API. For example, class posing has been deprecated, so you'll need to find another way to accomplish what you want.

Much of the updates are for the new 64-bit implementation of the runtime. Because they didn't have to worry about keeping backward compatibility, some of the changes could be more extensive. Some of the improvements are better performance, support for non-fragile instance variables, and zero-cost C++ compatible exceptions. Be aware, though, that some features that are only deprecated in the 32-bit implementation are removed completely from the 64-bit implementation, such as class posing. See "Objective-C Runtime Release Notes for Mac OS X v10.5" for full details.

New and Improved APIs

The tools and language are not the only area to see drastic improvements. There are new and improved APIs to go along with those changes.

Core Animation

Core Animation is one of those revolutionary APIs that will probably change the way nearly all Mac OS X applications look and behave. Core Animation takes the power of the graphics card and unleashes it to the masses. Typically, you would have to really dig deep into OpenGL to fully utilize this power, but Core Animation makes it much easier by providing an API for fluid animation, 2D rendering, and 3D projections.

Core Animation itself is a lower-level API, but the Cocoa Application framework provides direct support for it. At the simplest level, some properties of an `NSView` and `NSWindow` are now animatable. This means that the new value does not get set immediately, but takes place over time. For example, if you set the frame of a view like this, the frame gets updated right away:

```
[aView setFrame: newFrame];
```

If, instead, you use the `animator` proxy to set the frame, the frame will animate to the new value over time:

```
[[aView animator] setFrame: newFrame];
```

Besides the frame, other animatable properties include the alpha value, rotation, and background Core Image filters.

If you want to fully utilize the power of Core Animation, you'll need to dig deeper into the lower-level API and work with layers. Layers are similar to views, in that they contain drawable content at a specific geometry, but they are much more lightweight. For example, it's possible to have hundreds of layers being rendered at once. Layers can be transformed, rotated, and zipped around using animations. Unfortunately, this is too large of a large topic to provide many details here. I suggest you read the "Core Animation Programming Guide" and look at the sample code for more information.

Cocoa Application Framework

The Cocoa Application Framework, affectionately known as AppKit gets a nice update in Leopard. AppKit is now 64-bit capable, meaning you can write 64-bit GUI applications to work with large amounts of data. In previous versions of Mac OS X, only Unix command line applications could be 64-bit. The nice thing is that there are not separate 32-bit and 64-bit

CodeMeter: Ready for Leopard

Highest Security

- Vendor selectable secret and private key.
- Strong encryption algorithms with AES 128-bit and ECC 224-bit.
- Best-in-class tools for automatic protection (envelope, without source modification) for Win32, Win64, .NET, Java and MacOS X Universal (PPC, Intel).

Best Flexibility

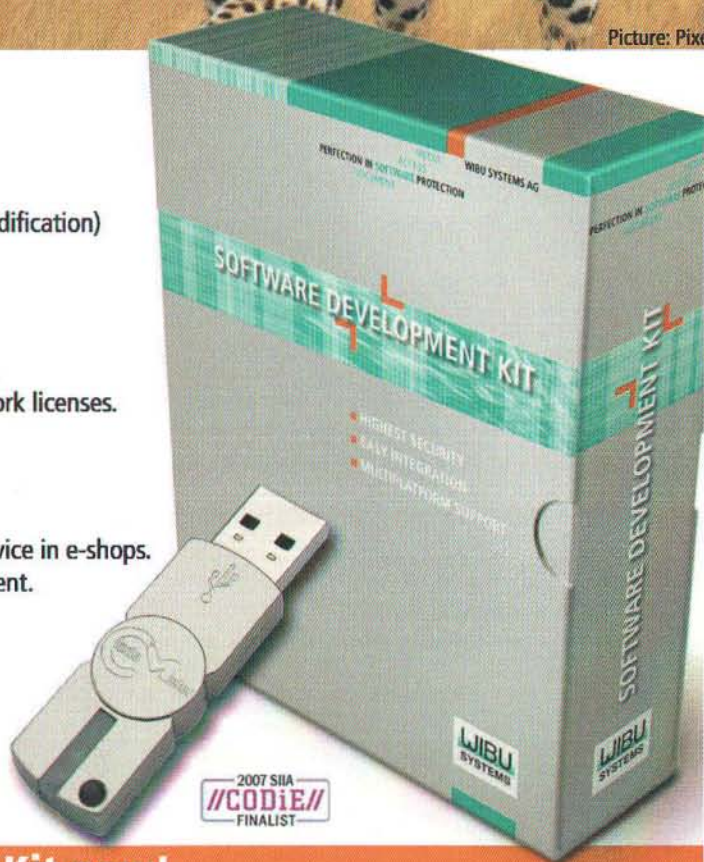
- More than 1000 independent licenses can be protected by one CM-Stick.
- One versatile hardware key for all license models including floating network licenses.
- Multi platform support including Windows, MacOS X and Linux.

New Distribution Channels

- License transfer by SOAP based CM-Talk or file based Field Activation Service in e-shops.
- Multiple-purpose, including protecting low cost software and digital content.

Unique End User Advantages

- First and smallest dongle with up to 2 Gbyte flash drive.
- No drivers necessary – can be used without administrator rights.
- CM Password Manager, secure virtual drive and secure login.



Order your Free Software Development Kit now!
Phone 1-800-6-GO-WIBU | order@wibu.us

Mac | Universal

WIBU-SYSTEMS submitted the CodeMeter Password Manager and the CodeMeter SDK for the **Apple Design Awards 2007.**

WIBU SYSTEMS

WIBU-SYSTEMS USA Inc.
2429 NW 197th Street
Shoreline, WA 98177, USA
www.wibu.com
info@wibu.us
Tel: 1.800.6.GO.WIBU
1.206.546.4891
Fax: 1.206.237.2644

versions of Leopard. 32- and 64-bit applications can be run side-by-side on the same machine, and you can even include both 32- and 64-bit versions of your binary in the same application bundle. Thus, you can ship a single binary to all your customers and not worry about what kind of machine they have.

There are quite a few new views and controls, too. Some of the standouts include:

`NSCollectionView` to provide an animatable list of views, like the pictures in iPhoto. The Library in Interface Builder is a nice example of `NSCollectionView`.

`NSGradient`, a class to easily create and display color gradients.

Native support for Heads Up Display (HUD) style windows. This is the dark gray, transparent window seen in many of the iLife applications, and it's available in Interface Builder.

Better Cocoa Bindings support with improvements to `NSTreeController` and `NSArrayController` and the addition of `NSDictionaryController`.

`NSRuleEditor` view to configure "rules" similar to Mail.

`NSPredicateEditor` view to create complex predicate expressions used in queries.

See the "Mac OS X v10.5 Developer Release Notes Cocoa Application Framework" document for all the changes to AppKit in Leopard.

Quartz Composer

Developers can now make their own Quartz Composer patches by subclassing `QCPlugIn`. This opens up the door for third party patches to make Quartz Composer even cooler than it already is. I am really looking forward to seeing what developers come up with, and how this is used inside of applications.

See the "Quartz Composer Custom Patch Programming Guide" for the full details on creating custom patches.

Other APIs

Other new and improved APIs and features include:

FSEvents to monitor file system updates.

Core Data 2.0, which includes support for versioning of managed object models and migration of data from one version to another.

Scripting Bridge to seamlessly interact with scriptable applications. For example, to get the current track in iTunes would look like:

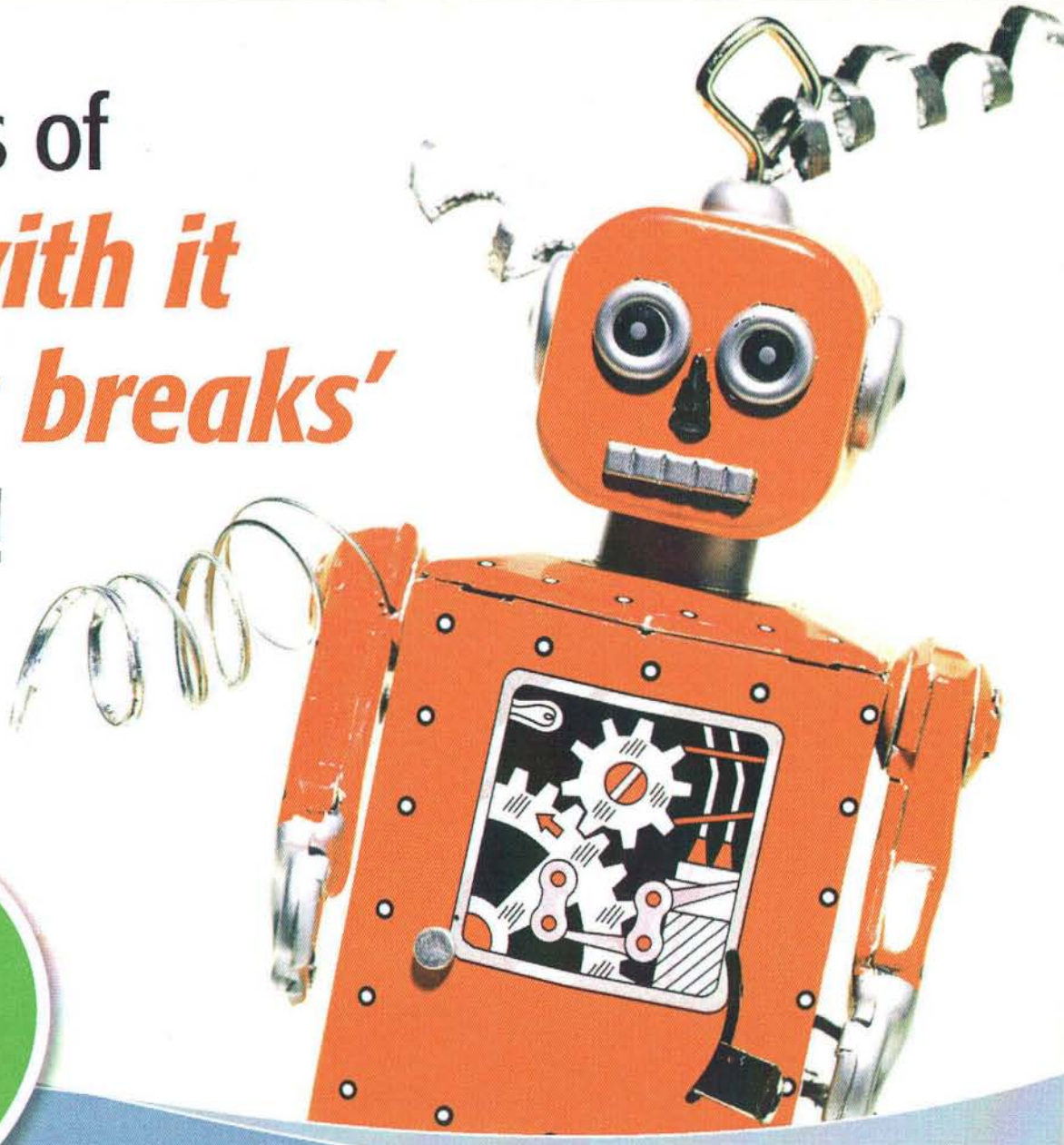
```
NSString * currentTrackName = [[iTunes currentTrack]
name];
```

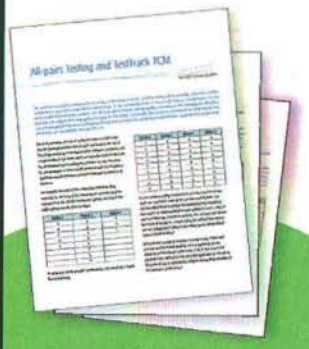
The Calendar Store framework to provide access to data from the iCal application.

Updates to the Instant Message framework to support injecting audio or video content into a running conference.

The Image Kit framework to support drawing of images and easier image manipulation.

The days of *'Play with it until it breaks'* are over!





Learn how to thoroughly test your applications in less time. Read our newest white paper: "All-pairs Testing and TestTrack TCM."

Download it today at www.seapine.com/allpairs1

Seapine

TestTrack® TCM

Software for test case planning, execution, and tracking

You can't ship with confidence if you don't have the tools in place to document, repeat, and quantify your testing effort. TestTrack TCM can help you thoroughly test your applications in less time.

In TestTrack TCM you have the tool you need to write and manage thousands of test cases, select sets of tests to run against builds, and process the pass/fail results using your development workflow.

With TestTrack TCM driving your QA process, you'll know what has been tested, what hasn't, and how much effort remains to ship a quality product. Deliver with the confidence only achieved with a well-planned testing effort.

- Ensure all steps are executed, and in the same order, for more consistent testing.
- Know instantly which test cases have been executed, what your coverage is, and how much testing remains.
- Track test case execution times to calculate how much time is required to test your applications.
- Streamline the QA > Fix > Re-test cycle by pushing test failures immediately into the defect management workflow.
- Cost-effectively implement an auditable quality assurance program.

Seapine ALM Solutions:

- **TestTrack Pro** Issue & Defect Management
- **TestTrack TCM** Test Case Planning & Tracking
- **Surround SCM** Configuration Management
- **QA Wizard Pro** Automated Functional Testing

Seapine Software

Download your fully functional evaluation software now at www.seapine.com/mactcm or call 1-888-683-6456.

The PubSub framework to support RSS and Atom.

Updates to QuickTime Kit to support audio and video capture.

And much, much more! See "What's New In Mac OS X" for the complete list.

Resolution Independence

LCD monitor technology is improving rapidly, cramming more and more pixels into an inch than ever before. Ultimately this means crisper images and text. Unfortunately, there is currently little OS and application support for these higher resolution monitors. If no scaling is done, then text becomes too small and unreadable. If the OS tries to scale the user interface, then graphics can look pixelated or show artifacts.

With Leopard, Apple is pushing developers to make their applications resolution independent. Resolution independent means that the graphics and text should look crisp and clear on all monitors, no matter their resolution. In practice this means updating your artwork so that it scales properly. You also need to update any custom drawing code to make sure it does not assume a specific screen resolution. You can test your application to see what it would look like at different resolutions using the Quartz Debug tool.

Apple is obviously preparing developers so it can start using these higher resolution monitors in their products at some point in the future. We don't know when this is, so it's better to start early and begin updating your applications when Leopard comes out. Consult the "Resolution Independence Guidelines" document for more information on this topic.

Other Languages

The final feature I'm going to highlight is the new support for scripting languages, such as Ruby and Python. It is now possible to write full-blown Cocoa applications using Ruby and Python instead of Objective-C. This has been possible on previous versions of Mac OS X by downloading third party bridge frameworks, but Apple is now including RubyCocoa and PyObjc as part of the standard installation. This means you no longer have to bundle these frameworks with your finished application. It also means that Xcode includes templates for Ruby- and Python-based applications out of the box. Finally, it means that Apple will be supporting any bugs and issues that crop up with these bridges.

To give you a small taste of what this means, here is a short example of subsclassng `NSView` in Ruby, taken from the DotView sample RubyCocoa application:

```
require 'osx/cocoa'

class DotView < OSX::NSView

  ib_outlets    :colorWell, :sizeSlider

  def awakeFromNib
    @color = OSX::NSColor.redColor
    @radius = 10.0
    @center = OSX::NSPoint.new(bounds.size.width / 2,
                  bounds.size.height / 2)
    @colorWell.setColor(@color)
    @sizeSlider.setFloatValue(@radius)
  end

  def drawRect (rect)
    OSX::NSColor.whiteColor.set
    OSX::NSRectFill(bounds)
    dot_rect = OSX::NSRect.new(@center.x - @radius,
@center.y - @radius,
                  2 * @radius, 2 * @radius)
    @color.set

OSX::NSBezierPath.bezierPathWithOvalInRect(dot_rect).fill
  end

  def isOpaque
    true
  end

  def mouseUp (event)
    @center = convertPoint(event.locationInWindow,
:fromView, nil)
    setNeedsDisplay true
  end

  def setColor (sender)
    @color = sender.color
    setNeedsDisplay true
  end
  ib_action :setColor

  def setRadius (sender)
    @radius = sender.floatValue
    setNeedsDisplay true
  end
  ib_action :setRadius

end
```

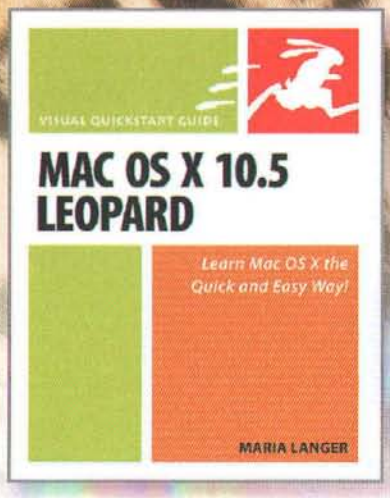
Apple is also making it easier for other Objective-C and Cocoa bridges by creating a new framework called Bridge Support.

Conclusion

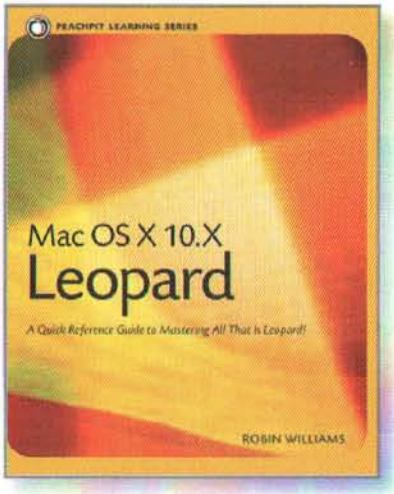
Well, that's a whirlwind introduction to Leopard. The developer has a lot to look forward to, and I think it is a very exciting release. Next month, we'll be back to our regular scheduled programming: learning the basics of Objective-C and programming for Mac OS X.

About The Author

Dave Dribin has been writing professional software for over eleven years. After five years programming embedded C in the telecom industry and a brief stint riding the Internet bubble, he decided to venture out on his own. Since 2001, he has been providing independent consulting services, and in 2006, he founded Bit Maki, Inc. Find out more at <http://www.bitmaki.com/> and <http://www.dribin.org/dave/>.

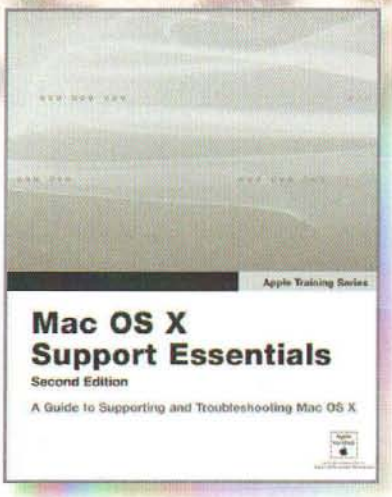


Mac OS X 10.5 Leopard: Visual QuickStart Guide
ISBN: 0321496000
Maria Langer
Available now!

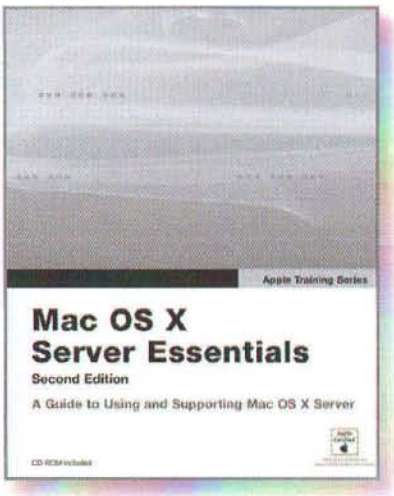


Mac OS X 10.5 Leopard: Peachpit Learning Series
ISBN: 0321502639
Robin Williams
Available now!

Automator for Mac OS X 10.5 Leopard: Visual QuickStart Guide
ISBN: 0321539354
Ben Waldie
December 2007



Apple Training Series: Mac OS X Support Essentials, Second Edition
ISBN: 0321489810
December 2007



Apple Training Series: Mac OS X Server Essentials, Second Edition
ISBN: 0321496604
December 2007

Are you ready for Leopard?

Learn the ins and outs of all the new features in Mac OS X 10.5 Leopard, including Time Machine, Spaces, Quick Look, Automator, and more. Whether you're an experienced Macintosh user ready to hone your trouble-shooting skills, or you're leading a group of beginners who need help getting started, pick up a Peachpit book today to get the most out of the powerful features in Leopard.

BUY NOW AND SAVE!

Purchase or preorder your copy today and **save 35%** off the list price, plus free domestic shipping. Simply go to **www.peachpit.com/leopard** and enter coupon code **PP-LEOPARD** when you reach the checkout page.

Peachpit

Apple Certified

Changing Spots

New ways of manipulating Directory Services in Leopard

By Philip Rinehart, Yale University

MacEnterprise.org
Mac OS X enterprise deployment project

User Account Changes

Occasionally, questions about user accounts and how to maintain them appear on the Macenterprise list. Leopard changes the landscape, as NetInfo no longer exists. What does this mean practically? All user account information is now stored as flat text files. No more messing around with command line utilities like nicl, nidump, etc. All access is now accomplished using the command line directory service utility, dscl. Let's take a look at how it works.

DSCL

Dscl, **D**irectory **S**ervice **C**ommand **L**ine, utility was originally introduced in Tiger. With the death of NetInfo, it is the new way of manipulating user accounts. Also, astute users may note that NetInfo Manager is now completely gone, so any manipulation of NetInfo attributes must be accomplished by using dscl. Directory Service attributes can be changed, appended or deleted. Let's take a very basic example.

```
dscl . -read /Users/myuser
```

This example operates on the local node by using the period, and returns all of the attributes for myuser. The command returns a list of all of the values that would have been seen in NetInfo Manager. They are printed out as a single line for each value. While interesting, it only begins to tap dscl for its true power. Here's a second example, listing all the users on the local system, as well as their UniqueID values (UID).

```
dscl /Local/Default -list /Users UniqueID
```

This command is a really quick way to list any attribute of any user that is stored in the local Directory Services store. Notice a slight difference in this command? Instead of using a period, the full node is specified, in this case, the local database, /Local/Default. Let's step back just a second. Since NetInfo is gone, where is all the information? Here is the complete path:

```
/var/db/dslocal/nodes/Default/Users
```

Explore the contents of the directory, notice how everything is a plist? One of the decisions made when moving away from NetInfo is that all of the information is now stored in xml plist format in the above directory. As an interesting side effect, any properly formatted plist that is added to the user will now appear on the system as a valid user. Returning to our UID example, now that the UID is known for any user, it is a pretty simple operation to change a UID on the fly. Back to dscl:

```
dscl /Local/Default -create /Users/myuser UniqueID 503
```

This command takes the current UniqueID value for myuser and overwrites or it with the new value. Instead of the user's previous value for UniqueID, a new one has now been put in place. Note that use of the create option will completely overwrite any current value. If the value does not exist, it creates it in the plist.

PUMP IT UP

All of our example dscl commands will work in Tiger. Dscl in Leopard has been beefed up considerably. It now has the ability to read subkeys through the use of additional command line options. If you ever looked at a NetInfo record that contained mcx information, you know that mcx settings are typically sent to the client as a plist. Now that this information is stored in a flat plist with nested values, dscl needs a way to manipulate the data. New options have been added, readpl, readpli, and createpl, createpli. Unfortunately the syntax is difficult to master, as it requires a very specific format. Here's a somewhat simplified example for managed preferences.

```
dscl . -readpl /Users/myuser MCXSettings
mcx_application_data:com.apple.finder
```

Do you have what it takes **to survive at the edge?**

Engineered for extremes

Put the Seagate® EE25 Series™ hard drives to the test – and you might be surprised to find that the edge of endurance is farther than you thought.

When you build systems for industrial, automotive or military environments, you need to venture beyond the limits. Designed to power the world's toughest systems with up to 80GB, the EE25 Series drive supports 24x7 seamless operation in mission-critical applications, withstanding high vibration, intense humidity and extreme temperatures ranging from –30°C to +85°C.

Take it to the edge – register to qualify for a free EE25 Series evaluation drive

http://specials.seagate.com/everest/mac

Seagate

Note the syntax of the key, colons separate nested values. In this particular case, the managed preference key for the Finder is read. This example should give you a taste of how the command works, but getting the path exactly right can be a bit tricky. Fortunately, there is a way out of the weeds, with a new mcx options for dscl..

MCX!

One of the major complaints in previous versions of OS X was the inability to easily understand and manipulate managed preferences. Leopard is the first version of OS X that has options to help manage via script, or the command line. It has also been quite difficult to troubleshoot managed client preferences, and to truly understand what is going on when managed preferences are applied. Let's look at our friend dscl again, this time with an eye toward the options that were added to dscl. Here's a very simple example:

```
dscl . -readmcx /User/myuser
```

Note how the information is returned. Each managed preference is returned as a set of values with a consistent format. So for example, if a Finder preference was managed, the value might look like this:

```
App domain: com.apple.finder
Key: ComputerViewOptions_Arrangement
State: always
Value: None
```

Cool! The mcxread option is useful, but even more useful is the ability to set, import and export keys with dscl and its associated mcx commands. Imagine being able to set preferences from the command line from a client! A sample process could be:

```
dscl . -mcxexport /Users/myuser -o /tmp/export.plist
com.apple.finder
```

This command exports the managed client settings for myuser. The settings can then be altered in the exported file, export.plist with any text editor. Once finished editing, use this command to import the changed values:

```
dscl . -mcximport /Users/myuser -d /tmp/export.plist
```

One note about this command, the –d option deletes any keys that existed previously. It is equivalent to calling mcxdelete for **every** key found in the import file. There are many options available for command line managed preference manipulation of preferences, which are not documented in the manual page. So how can you find the proper options? Use the flag –mcxhelp.

```
dscl . -mcxhelp
```

This short command returns all of the options available, and is quite thorough in its description of how to use the command line options.

What if you only want to see what managed preferences are being applied? A new command for Leopard, mcxquery has been added. It can be called directly to present all of the options for any known user, group or machine. Here's how:

```
mcxquery -user myuser -group mygroup -computer
mycomputer
```

This command returns a list of all managed preferences for all three options. Additionally, it specifies exactly **which** domain the management is being applied from. If it is a user management preference, it indicates the managed preference. Very useful! Now that Leopard is finally out, a whole new world of discovery awaits us. As always, see you on the lists!

About The Author

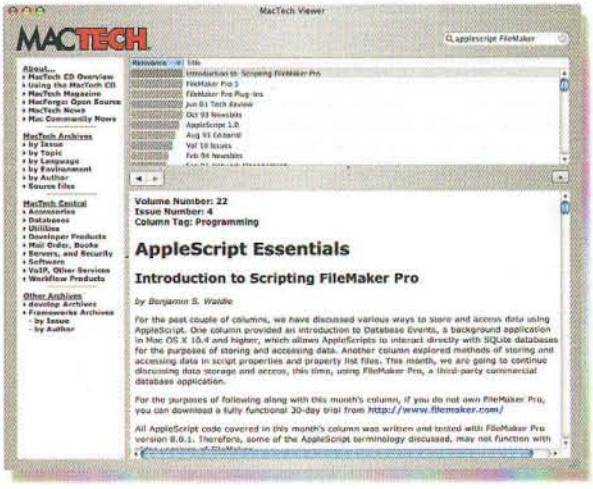
Philip Rinehart is co-chair of the steering committee leading the Mac OS X Enterprise Project (macenterprise.org) and is the Lead Mac Analyst at Yale University. He has been using Macintosh Computers since the days of the Macintosh SE, and Mac OS X since its Developer Preview Release. Before coming to Yale, he worked as a Unix system administrator for a dot-com company. He can be reached at: philip.rinehart@yale.edu.

The MacEnterprise project is a community of IT professionals sharing information and solutions to support Macs in an enterprise. We collaborate on the deployment, management, and integration of Mac OS X client and server computers into multi-platform computing environments.

It's ALL in there!

The MacTech CD - Volumes 1.01-23.06 is packed with more than ever before -- over 2900 articles from more than 260 issues (**1984 - June 2007**) written by over 840 experts, all 29 issues of Apple's develop, 21 issues of FrameWorks magazine, 100+ MB of source code, MacTech Viewer, working applications, full documentation, demos for techs, ***and more!***

Everything is displayed in the all new, very fast, very searchable ***MacTech Viewer!*** A new application that has been designed specifically with Techs in mind. Search quickly through *20 plus years* of great information provided by MacTech. Information to make your life easier.



Mac Universal

Requires Mac OS X v. 10.4.5 or later

Toll Free 877-MACTECH, Outside US/Canada: 805-494-9797 • http://www.mactech.com/cd/

Using the RADIUS Service in OS X 10.5

Aimed at Airports, but flexible under the hood

By Ben Greisler

Why RADIUS?

Controlling access to your wireless network has just become easier with the inclusion of RADIUS services in Leopard Server. Providing a central method for controlling user access to Airport access points (and other RADIUS capable devices), the Leopard implementation is Apple simple.

RADIUS Services in Leopard 10.5

Setting up RADIUS in Leopard server is very straightforward. You can either use the wizard or manually set it up manually. Normally I would say that using a wizard is the easy method, but in this case even the manual method is very simple to follow. Regardless of the method, you will end up with a working set up.

In my test environment I set up an Airport Extreme (round) and configured it to bridge. This has nothing to do with the RADIUS setup, but will be similar to how other will be using it. I had a working OD master on my network and I set up a group of users authorized to connect to the wireless network. I then bound my Leopard server to the OD and checked that I could recognize OD users from the Leopard machine. The RADIUS services can be set using SACL's (Service ACL's) making it very flexible from an authorization standpoint. Once that was set I was able to start the RADIUS configuration.

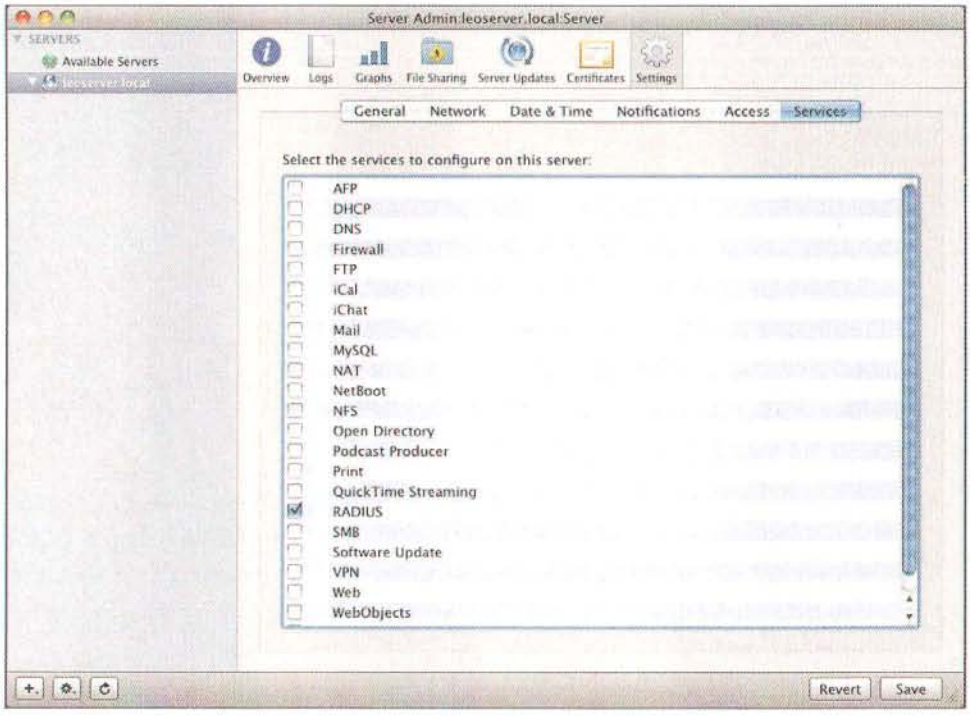


Fig. 1. Activating the RADIUS service in Server Admin

Wizard Method:

Open Server Admin and activate the RADIUS service in Settings/Services. This will make the RADIUS service available in the services list.

In the RADIUS service Overview pane, click the "Configure RADIUS Service..." button.

Revolutionize your Lifecycle Management Strategy with LANrev!

True Cross-Platform Lifecycle Management has arrived!
Do you need to manage PCs with a Mac? Macs with a PC?
LANrev is the first client management solution
with true cross platform functionality
for today's heterogeneous client environments.

Disk Imaging
Asset Inventory
Software Distribution
License Management
Patch Management
Remote Management
Role-Based Administration
Change and Configuration Management
LANrev TheftTrack Client Recovery

LANrev is a scalable multi-platform solution, which can operate with equal efficiency on a single server deployment in the small to medium enterprise, or in a distributed architecture, managing thousands of desktops in a large enterprise.

LANrev's modern, modular platform allows seamless integration into other desktop management tools such as Microsoft's SMS, enabling your organization to leverage previous investments while maximizing current and future efficiencies through the use of innovative new technology such as LANrev. The best part? You can take advantage of your current hardware infrastructure. Unlike the other guys, LANrev does not have massive, dedicated server requirements and you won't need a team of overpaid consultants to deploy and use LANrev.

See for yourself how you can increase your IT efficiency and performance while lowering your cost by using LANrev in your computing environment. Schedule a free web demo and trial license by calling (214) 459-0136 or visiting www.lanrev.com.

LANrev

Mac Universal for windows

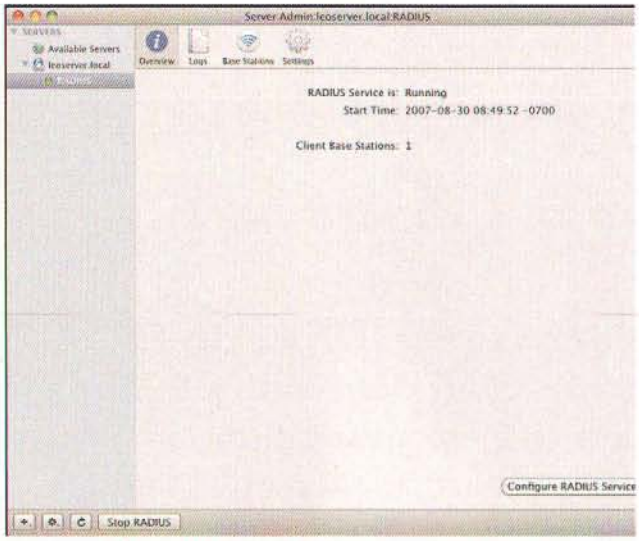


Fig. 2. RADIUS Service Overview pane

The first page lets you chose what certificate you want to use. You can pick an existing certificate or create a self signed certificate.

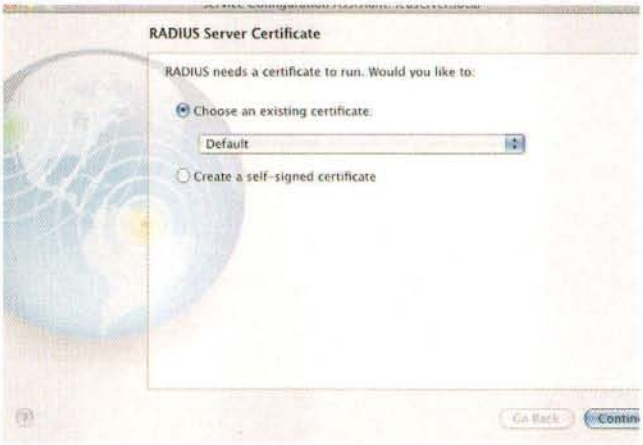


Fig. 3. Pick your certificate or make your own

With your Airports up and running, you should see them in the Add Base Stations page. Pick the ones you want to use RADIUS with and fill in the Airports administrative password, then click "Add." Once the base stations have populated the Selected Base Stations window, click "Continue."

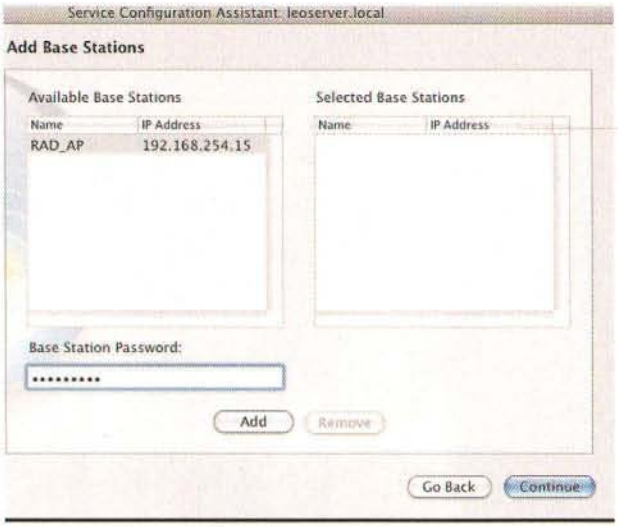


Fig. 4. Select the Airports you want authorized via RADIUS

The next window allows you set up who will be authorized to connect to the wireless network. We had set up an Open Directory group named "radiususers" in this example and selected it.

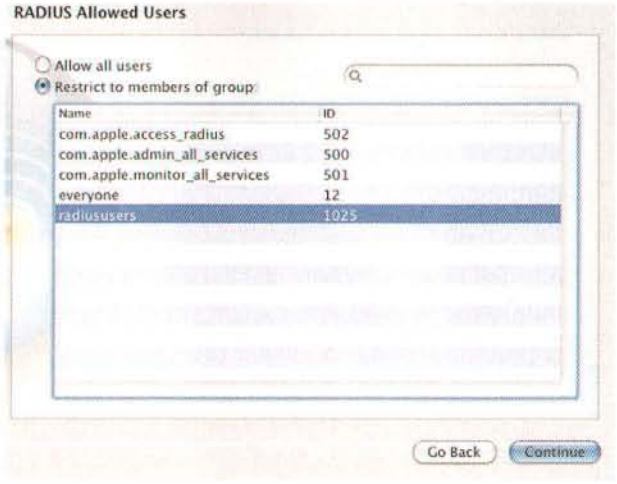


Fig. 5. Pick your group to limit users or allow all users

In the next windows, much like the final steps of configuring an OSX Server, you are given the opportunity to do a final check of your RADIUS configuration.


save the date!
january 14-18, 2008
the moscone center
san francisco, ca
Macworld
Conference & Expo®
macworldexpo.com

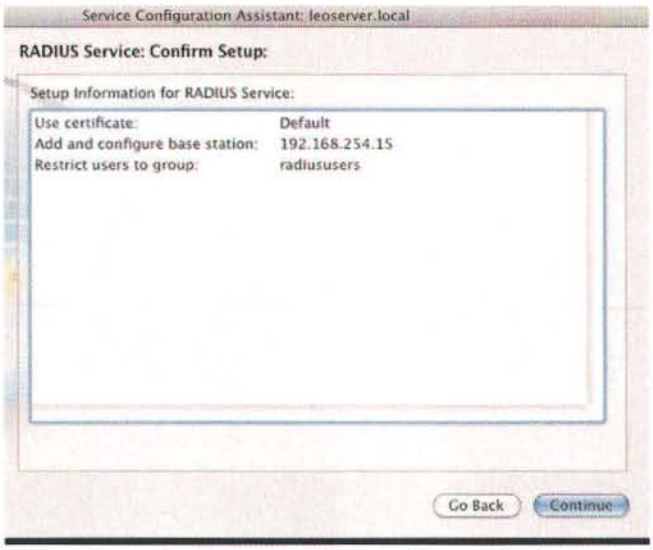


Fig. 6. Check your settings and commit them

The final window simply announces the successful configuration of RADIUS services.

Fig. 7. Success! Start logging in!

Manual Method:

Open Server Admin and activate the RADIUS service in Settings/Services. This will make the RADIUS service available in the services list.

In the RADIUS service Settings pane, we need to configure our certificates. If you choose a custom configuration you will be presented with you choices of where the certificates reside and a quick link to the Certificate Management page.

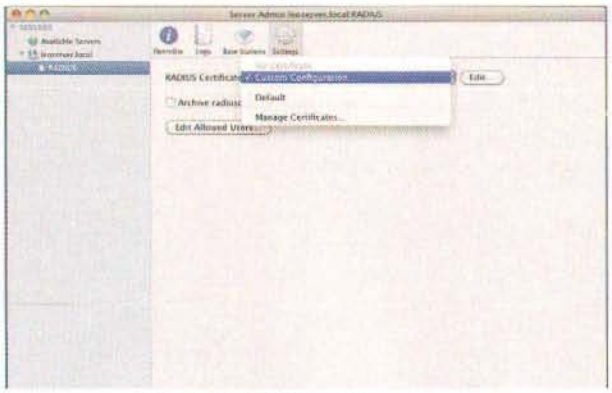


Fig. 8. RADIUS settings page

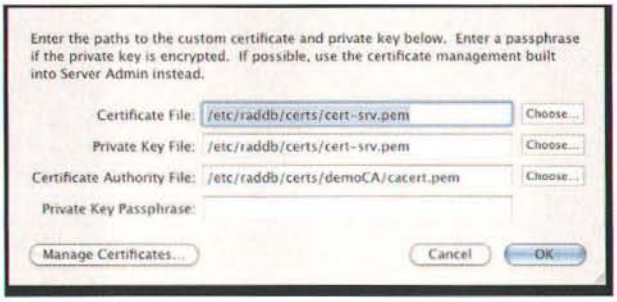


Fig. 9. Certificate locations

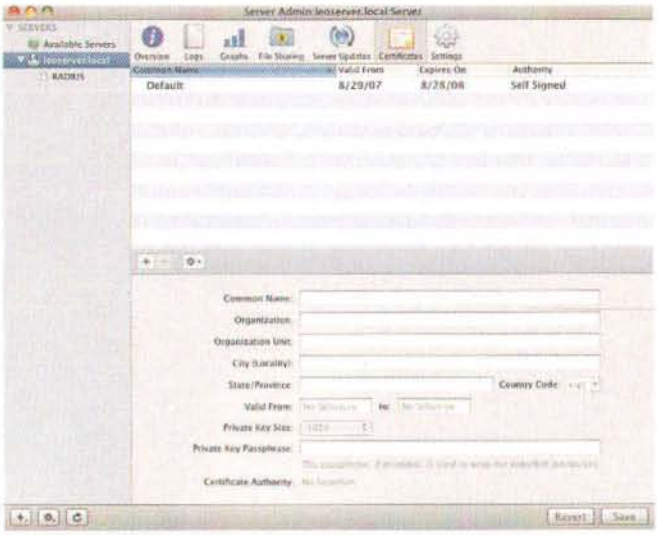


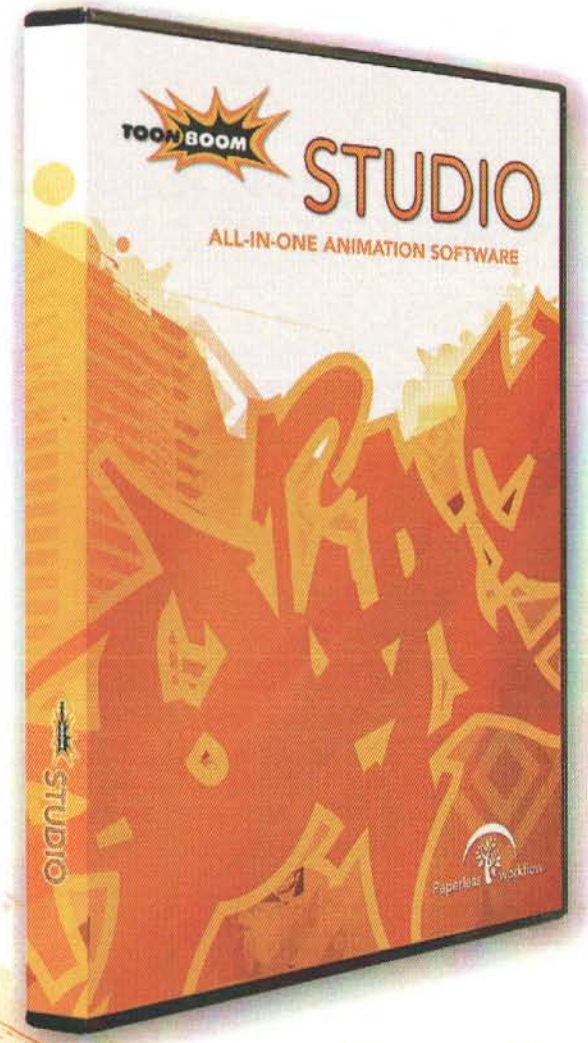
Fig. 10. Certificate management

The RADIUS settings page has an "Edit Allowed Users..." button bringing you to the SACL's page where you can define which groups can be authorized.

ANIMATE WITH EASE!

TOON BOOM STUDIO

ALL-IN-ONE ANIMATION SOFTWARE



The only start to finish animation software toolset designed for creative and professional users.

- Draw, scan or import artwork into system
- Paint artwork automatically
- Lip-sync mouth movements to voice tracks
- Animate in conjunction with sound, effects
- Access libraries to use and reuse content
- Create motion and camera moves in 3D
- Apply special effects
- View animation in real time
- Publish to TV, HDTV, video, web (SWF, QT)

Teacher's guide and animation lessons available

toonboom.com

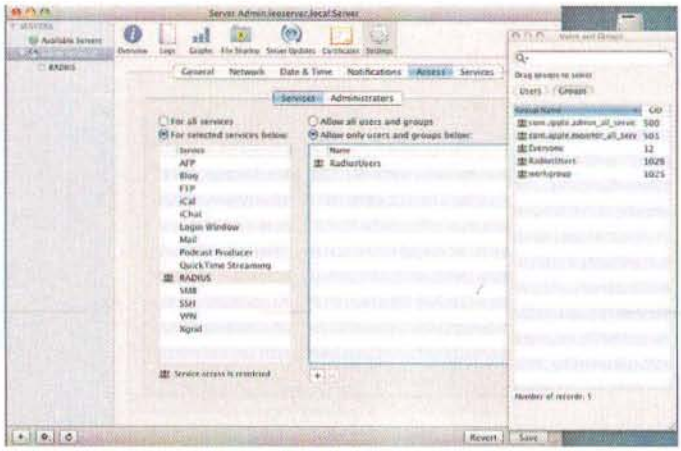


Fig. 11. When defining an authorized group, you can use any method of building the groups for maximum utility such as nesting

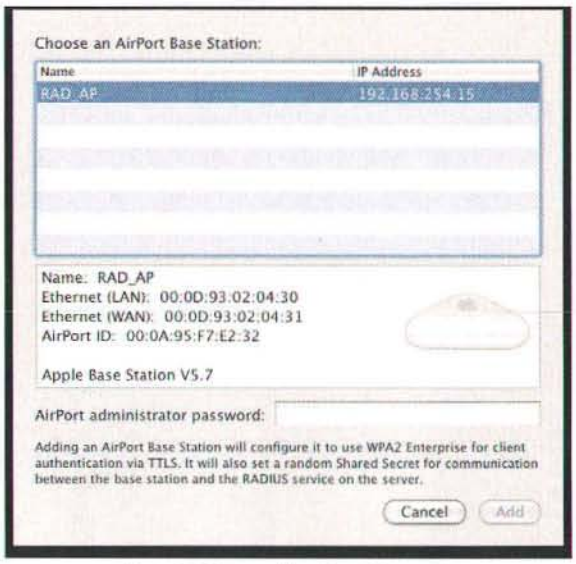


Fig. 13. Browsing for Airports

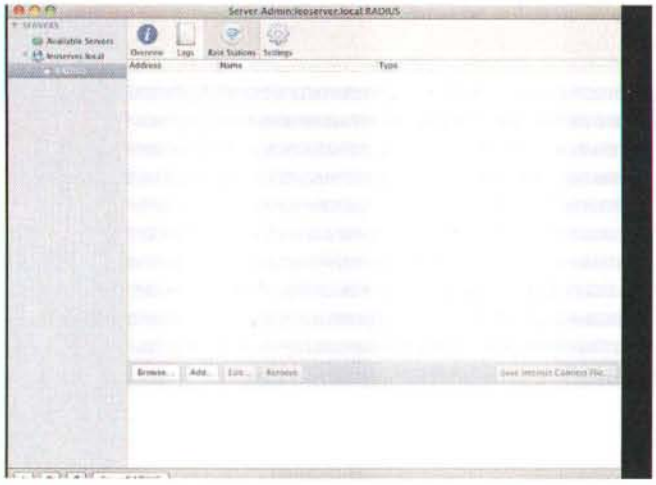


Fig. 12. The Base Stations window gives you the tools to manage your access points

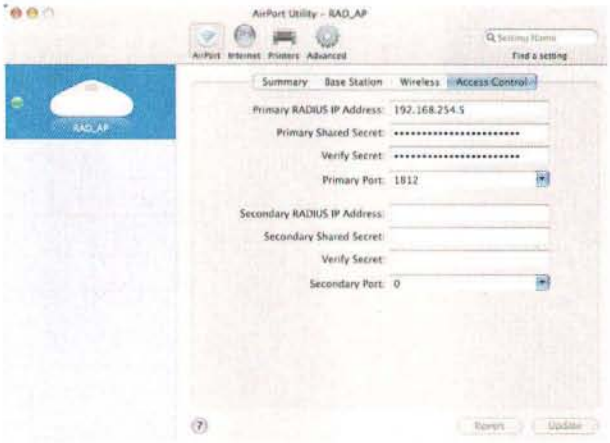
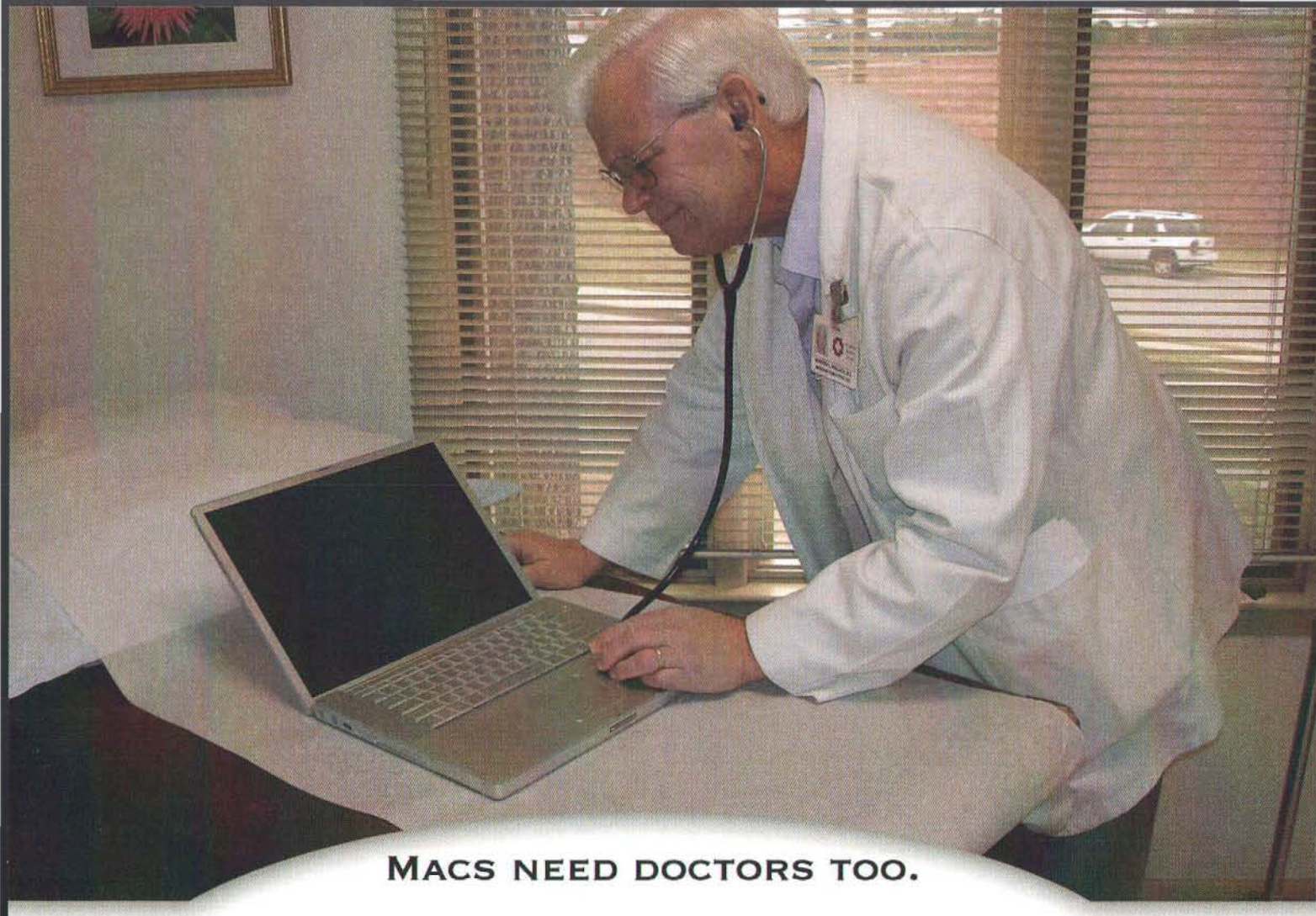


Fig. 14. If we now look at the Aiport in the AirPort Utility we can see that the RADIUS configuration has been set for us

In the Base Stations window you can browse for Airports or add them manually

When browsing for Airports, you simply pick the devices you want, type in the administrator password for the device and the RADIUS service does the rest. It will communicate with the Airport, define a random shared secret and restart the Airport.

If we pick the "Add" button, we are able to directly configure an Airport. It won't be as easy, but it is necessary for Airports not visible while browsing. You will also need to manually configure the Airport in the AirPort Utility with the shared secret. You could also set up non-Airport devices using this method.



MACS NEED DOCTORS TOO.

PowerbookMedic.com

Your Mac — Our Patient

HARD DRIVES

ADD MORE SPACE FOR MUSIC, MOVIES, & ALL OF YOUR FILES.

HARD DRIVES FROM $2.99

BATTERIES

LONGER LIFE. CHEAP PRICE.

BATTERIES FROM $99.95

MEMORY

SPEED UP THAT SLOW MAC.

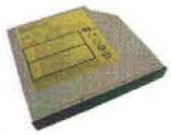
MEMORY FROM $19.95

KEYBOARD / KEYS

REPLACE THOSE MISSING KEYS.

REPLACEMENTS FROM $8.95

DRIVE UPGRADES



BURN CDS AND DVDS! UPGRADE YOUR OPTICAL DRIVE.

UPGRADE FOR $119.95

PARTS, UPGRADES, AND REPAIRS FOR YOUR MAC.

WWW.POWERBOOKMEDIC.COM

Fig. 15. Configuring the Airport manually

To ease the rollout to Mac clients, you can export an Internet Connect plist. Click on the "Save Internet Connect File" button.

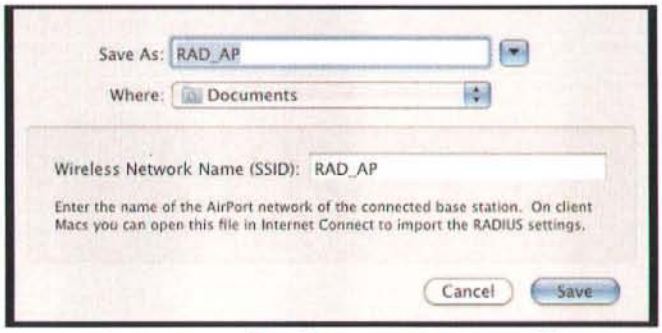


Fig. 16. Exporting the Internet Connect file

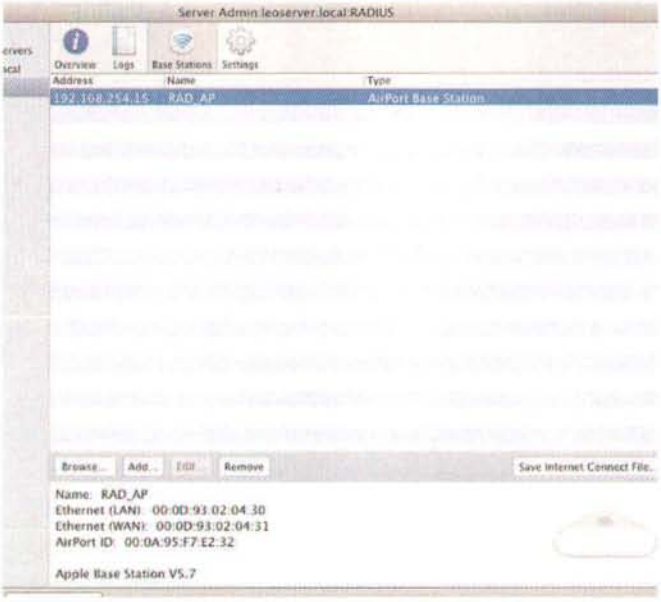


Fig. 17. With everything else configured we can get our clients connecting

Fig. 18. The user experience when connecting to a RADIUS controlled Airport

Command Line:

RADIUS in Leopard is based on FreeRADIUS and is configurable via command line. Visit the man page for details:

```
leoserver:~ leoadmin$ man radiusd
```

Take particular note of the ordering of the radius.conf contents if you decide to modify it directly. A change of ordering can make the server not work any more. This is noted in the man page and is worth repeating.

If you want to examine the files that make RADIUS work, go to /etc/raddb. Logs live at /var/log/radius/radius.log.

Conclusion

RADIUS services in Leopard server is a great addition to the set of tool Apple has given us. While Apple's implementation is really aimed at Apple Airport users, it should also work with other devices.

MT

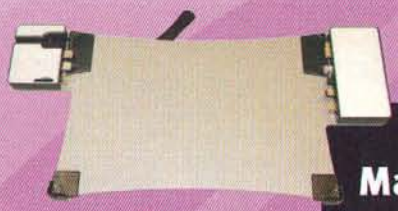
About The Author

Ben has been everything from a Mac user to CTO of one of the leading Macintosh professional services firms. Besides writing an occasional article for MacTech, you can find him presenting at Macworld (including a session called "DNS: Demystified, co-presented with Doug Hanley) or consulting with clients around the world. You can reach him at ben@greisler.org.

BookEndz®

DOCKING STATIONS FOR APPLE COMPUTERS

Convert your MacBook Pro®, MacBook®, PowerBook®, or iBook®
into a desktop in seconds without misplacing cables or damaging connectors.



17" MacBook® Pro Docking Station

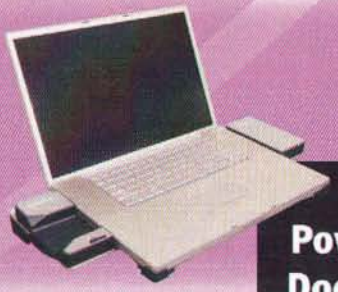
- Aluminum Plate helps in cooling of notebook
- Connectors are routed to rear of Dock
- MagSafe® Housing prevents accidental disconnect

15.4" MacBook® Pro Docking Station

13.3" MacBook® Docking Station

- Also available in Black
- Built in DVI/VGA full size
- 5 USB 2.0 compliant ports
- Gigabit Ethernet RJ45

(4 port powered or unpowered hub, AC/DC adapter included)



17" PowerBook® G4 Docking Station

12" PowerBook® Docking Station

15" PowerBook® G4 Docking Station

iBook® G3/G4 Models Docking Station

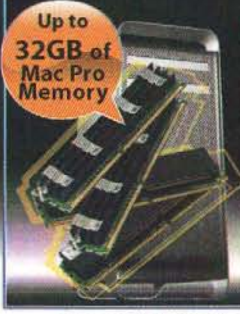
Visit our website for latest product announcement www.BookEndzdocks.com

BookEndz®
Manufactured by Olympic Controls
1250 Crispin Drive • Elgin, Illinois 60123 • USA
Phone: 847-742-3566 • Fax: 847-742-5686 • Toll Free: 888-622-1199
E-mail: Sales@BookEndzdocks.com

MEMORY UPGRADES

www.macsales.com/memory

Providing the Right Memory for Apple® Computers



OWC Mac Pro 8-core/Quad-core Memory
1GB, 2GB, 4GB Matched Sets for up to 32.0GB

Mac Pro Upgrades from **$65.56** per GB

OWC Mac Pro 8-core/Quad-core Memory
Fully tested and certified to meet or exceed all Apple specifications for use in Mac Pro for a lifetime of performance. OWC Memory includes the use of actual Apple-Qualified Heat-Spreaders to ensure you can utilize these models with full confidence.

Detail of Apple-Qualified OWC heat spreaders for Mac Pro Quad Xeon

We have the right memory for your Mac®
OWC has memory for nearly every Mac including G3, G4, and G5!

Cash Back!
OWC offers memory trade in rebates! See *www.macsales.com* for details.

All memory from OWC meets or *exceeds* Apple's specifications and is backed by a 30 day money-back guarantee and LIFETIME Advance Replacement Warranty.

HARD DRIVE UPGRADES

www.macsales.com/harddrive

2.5" SATA
up to 250GB!
120GB for **$82.99**

3.5" SATA I/II
up to 1.0TB!
300GB to 1.0TB from **$64.95**

3.5" IDE/ATA
80GB to 750GB from **$52.99**
250GB from **$69.99**

2.5" IDE/ATA
NOW up to 250GB!
100GB for **$57.99**

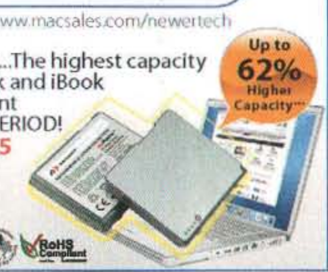
LAPTOP BATTERIES AND CHARGERS

newertechnology

www.macsales.com/newertech

NewerTech® 2-Bay Battery Charger+Conditioner.
Delivers the longest runtimes and extends lifetime from PowerBook®, iBook®, MacBook®, and MacBook Pro batteries.
from **$149.95**

NuPower®...The highest capacity PowerBook and iBook replacement batteries PERIOD!
from **$99.95**



iPOD ACCESSORIES AND BATTERIES

newertechnology

Mod Your Pod

www.macsales.com/ipod

TRAFFICJamz™
FM Transmitter + iPod® Charger.
Play your iPod through your car stereo while charging it.
Only **$34.95**

ViDEO+™
Give your 5G iPod up to 16 hours of video or 80 hours of music play.
Only **$49.99**

NuPower
from **$19.99**

OWC Mercury Elite™ Pro RAID Subsystem - eSATA/FW800/FW400/USB2 Solutions
For DVR or SMB customers whom have large data storage/backup requirements.

- Hardware RAID Design, supports RAID 0,1,3,5,6 & Hot Spare Disk.
- Supports 5 Hot-Swap Trays, RAID, and multiple interfaces.
- Online Data Auto-Rebuilding.
- Greater capacity up to 2.0TB per volume set.
- Includes an independent microprocessor design which will not affect system performance.

SUPERDRIVES®

www.macsales.com/superdrives

Internal SuperDrives for Desktops/Towers from **$34.95**

Internal SuperDrives for Apple Laptops from **$109.99**

External SuperDrives USB 2.0/FireWire from **$79.99**

Other World Computing
Serving the Mac Universe Since 1988

Since 1988

Memory for Apple's Intel® Macs
Upgrade Apple MacBook, MacBook Pro, iMac*, and Mac mini Core Duo/Solo, Core 2 Duo, and Core 2 Extreme. Now up to 4.0GB Available!†

Featured! iMac & MacBook Memory

1.0GB from **$33.99**
2.0GB from **$97.99**
3.0GB Upgrade Kit* **$129.99**
4.0GB Upgrade Kit** **$195.00****

"That's a great deal!" -Dave Hamilton, Mac Observer

*For all speed Core 2 Duo Equipped Models **For 2.2GHz and 2.4GHz Core 2 Duo models †For Apple MacBook Pro Core 2 Duo 2.2Ghz and 2.4Ghz models only

VISIT OWC AT BOOTH #2218

Macworld Conference & Expo

January 14-18, 2008 • The Moscone Center • San Francisco

Hard Drive Controller Cards
from $17.99

SOFTWARE

www.macsales.com/osx

PROSOFT engineering inc.

Prosoft Data Rescue II
The best Mac data recovery software on the market for recovering files from a problem hard drive.

Only **$59.99**

Prosoft Drive Genius v1.5
Maintain, manage, optimize, and repair your hard drives. Fully compatible with G3/G4/G5 and Intel Macs.

Only **$75.99**

GUARDIAN MAXIMUS

www.macsales.com/storage

newertechnology

Guardian MAXimus

"LIVE ACTIVITY" Backup

- FireWire® 800/400+USB 2.0 Hardware RAID-1 (Mirrored) Data Redundant Solution
- Up to 1.0TB of storage capacity
- Provides "live activity" backup of critical data
- Plug and Play simplicity on any Mac or PC

250GB to 1.0TB from **$319.99**
500GB only **$419.99**

Add an external hard drive rack solution from OWC, and immediately open the door to more storage and faster performance. Most importantly, you will get the security and peace of mind of having an award winning, precision backup.

- RAID 0 & JBOD mode for customizable configuration.
- Host Interface: 2 set of Firewire (800) 1394b & USB 2.0.
- 2 sets of dedicated Firewire channel for ultimate performance.
- Supports 3 Different Configuration Modes.
- Up to 4.0TB in storage capacity.

OWC Mercury Elite™ Pro RAID Rack Mount SATA-II Storage Enclosure

Ideal for applications requiring high data throughput, availability, and flexibility in configuration.

www.macsales.com/raid

EXTERNAL STORAGE SOLUTIONS (FIREWIRE, USB, & eSATA)

www.macsales.com/storage

Performance Plug and Play FireWire 800/400, USB 2.0, & eSATA

OWC Mercury On-The-Go
FireWire, FireWire +USB2.0, USB2.0, USB2.0+eSATA
Solutions up to **250GB**
from **$87.99**

OWC Mercury Elite-AL Pro
'Quad Interface'
FW800/FW400/USB2.0
250GB to 1.0TB (1000GB)
from $175.00
'Triple' FW800/400+USB2.0
80GB to 750GB
from $129.99

OWC Neptune
FW400
from **$91.99**
250GB for **$114.99**
500GB for **$167.99**

OWC Mercury Elite-AL Pro
Performance RAID Solutions **320GB to 2.0TB (2000GB)**
from **$229.99**

$100 Software Included

Prosoft Engineering® Data Backup 3, NovaStor NovaBACKUP®, and Intech® SpeedTools Utilities™

...a $200 Retail Value FREE! *Included with all OWC Mercury Elite, Mercury On-the-Go, OWC Neptune, and NewerTech miniStack® solutions.*

Build your own drive

FireWire/USB2.0/eSATA

3.5" Single & Dual Kits from **$44.95**

2.5" Portable Kits from **$29.95**

800.275.4576
www.macsales.com

BizRate 2007

No Matter The Mac We've Got You Covered!

From 60x, G3, G4, G5 to Intel Desktops, Towers, Laptops, and All-In One, OWC has the Goods to Maximize Your Mac.

USB 2.0 UNIVERSAL DRIVE ADAPTER

newertechnology

Macworld

The Swiss Army Knife of Disk Connectivity

Gives New Life to Old Drives!
Connect any 2.5", 3.5", 5.25" IDE and SATA device to your computer through your USB 2.0 port. Easily transfer files from computer or notebook, back up files, or store large file archives on hard drives. Only **$29.95**

POWER MAC G4 PROCESSOR UPGRADES

www.macsales.com/faster

Put a faster processor in your Mac for up to 18X the performance.
from **$199.00**

G4/1.2GHz for **$199.00**
1.5GHz **$259.99**
2.0GHz **$399.99**
Dual 1.8GHz from **$549.99**

Affordable NAS

Integrating FreeNAS appliance software into a Mac environment

by Robert Staehl

What This Article Will Cover

The purpose of this article is to introduce the user to the potential of integrating a NAS device – an open source, Mac-friendly (AFP serving) NAS appliance – into a network environment. The article will focus on getting a FreeNAS device configured and running on a local network. FreeNAS includes several advanced features such LDAP and AD support that I do not cover in detail. This article will cover installation and configuration to a point allowing network access to the data stored on the device.

Why NAS?

NAS, or Network Attached Storage, is a means of sharing data and files via a network. A typical scenario is the use of a NAS device within a work group as a way to store information which allows multiple protocols access, such as NFS, SMB and AFP. As an example, in our office we work with applications that are specific to Mac or Windows systems that need to share common assets. A NAS is a logical solution that allows for the storage of such assets without the need to maintain a full file server. Even for a home network, a NAS appliance can be a rational location for the central storage of music and video libraries.

Why FreeNAS?

My personal experience started by attempting to use a boxed NAS solution available either on line or from the local computer shop. I found out the hard way that the major brands, while claiming to support OSX, integrate an older version of AFP that does not support long filenames. Being cheap to a fault and having extra Intel PC's hanging around my office, I decided to try out the Appliance Software route. There are other widely used NAS solutions that include 1u or 2u chassis preconfigured that only require adding drives: ReadyNAS looks to be a great solution. My goal was to build a cheap, massive (more than 4 drive) solution using mostly left over parts hanging around. To achieve this goal, FreeNAS fit my needs perfectly. An added benefit of using FreeNAS is that it is open source and can potentially be modified or expanded upon as my business evolves. A proprietary solution inherently lacks this flexibility.

Evolution of Appliance Software

Many specialized tasks have been distilled to a self-contained package of tools bundled with a compact version of an operating system (http://en.wikipedia.org/wiki/Software_Appliance). Appliance Software is perfect for dedicated tasks such as routers, mail servers, database applications, NAS systems and almost any other isolated task imaginable. There are various resources on the Internet for finding Appliance software but one of my favorites is the list hosted by VMWare (currently located at the following URL: http://www.vmware.com/vmtn/appliances/directory/). Appliance software can run the gamut from free open source, to commercial packages that can cost thousands of dollars. FreeNAS is an appliance software package. It is an open source initiative based on the FreeBSD kernel.

Getting FreeNAS

FreeNAS is available on the FreeNAS.org website. In the downloads section of site, there are several choices for versions and formats. Before downloading, it is best to consider how the appliance will be run. For my personal needs, a dedicated PC with a 3ware RAID card and 8 SATA drives is the plan. To save space in the case and avoid the added power demands of a dedicated hard drive for the OS, I've opted to run the operating system from a USB jump/thumb drive. The best option for this type of a setup is to obtain the .iso of the latest build of

FreeNAS. Normally for a production environment I'd recommend only downloading the latest 'Stable' version of an open source package. But since FreeNAS is currently still in pre-release (there is no 1.0 version yet), beta is the only option. As with any new technology introduction, standard disclaimers apply: test, test, and test prior to deployment to production!

Target Box

My target machine is a standard Intel based PC with a 3ware 8500 series 8-port SATA card. Except for the RAID card, the PC is built with generic components (including motherboard, processor, ram and NIC). The power supply is 500 watt with lots of drive connectors. I found a 'RaidMax' chassis at a local Fry's store with lots of room for drives and good cooling. For purposes of installing the FreeNAS software, I included a CD-ROM drive in the machine build. The only potential 'gotcha' is the system BIOS will need to support booting from USB if the OS is to be installed on the jump/thumb drive.

Installation and Configuration

Burn a CD from the ISO image you downloaded from FreeNAS.org, and boot the PC from it - set the BIOS of the PC to boot from the CD if you must. The PC should boot with a FreeBSD kernel and detect the hardware connected. After the completion of the boot process, there should be a console setup menu with 9 choices. Insert the target USB jump/thumb drive, the console should return some messages that the drive has been detected. Since we plan to install to the USB drive we select item 9 in the list ('Install to a hard drive/memory drive/USB pen, etc.'). After some warning about erasing data on target drives there should be a submenu of 4 choices. Sub-item 1 will install the FreeNAS using all available space on the USB drive. After selecting a single or two partition install (Sub-item 2), the installer will ask for the location of the source data (the CD-ROM). In my case, device 'acd0' is my source drive. So, I type 'acd0' at the prompt (without the quotes). After hitting return, the installer lists all the potential target drives for installation. In my case, again, device 'da0' is the USB drive, so I type that name at the prompt. After a few minutes, the installation should complete and provide instructions for rebooting the system. The CD can also now be removed. The complete FreeNAS installation consumes less than 40 megabytes so it should fit on almost any spare or unused USB drive.

On the next reboot, the BIOS will need to be tweaked again to set the USB drive as the primary device. After proper configuration of the BIOS, the machine should boot into the FreeNAS OS from the USB drive (if your machine supports doing so). Otherwise, you can always install on a hard drive. When the system is booted, the original 9-item menu should present itself again. The first order of business will be to setup the network. Item 1 in the menu is used to Assign Interfaces. Meaning that if the system will have more than one NIC, this is where they could be set to WAN and LAN. For my purposes,

the NAS device will only be used on the local network so I only need to configure a single NIC as the LAN. After assignment of the interface, the system may need to be rebooted. This should be the last reboot until the system needs to be taken offline for maintenance.

We should be back to the 9-item menu. This time select item '2', this is used to assign an IP address to the device. I like to manually assign an IP rather than use DHCP, that way, I can set my local DNS to point directly to the box and assign a name like nas.mydomain.com to the IP. Follow the on-screen instructions to assign an IP and subnet to the LAN. The last task at the console is assigning a webGUI password (item 3). Set an appropriate password by following the on screen instructions.

WebGUI

It's now safe to leave the console and return to a machine running a friendlier interface. From another machine on the same subnet, launch a browser and connect to the IP assigned to the FreeNAS box. An authentication window should be presented. The username is 'admin', the password should be whatever you assigned in the last step at the console. If everything worked correctly, there should be an interface that appears similar to that of figure 1.

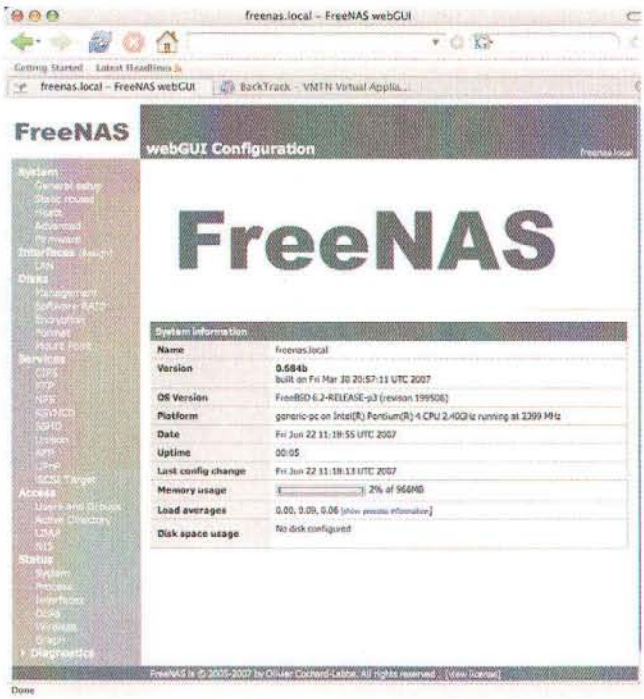


Figure 1 – The FreeNAS web console

From this GUI, the appliance can now be configured for the installed drives and how the users on the network will see the drives. The first order of business is to show the FreeNAS OS the drives to share. One of the reasons I like 3ware RAID cards is that support for them is built into the kernel of FreeBSD (and hence into FreeNAS). 3ware has a BIOS-level interface that allows the drives to be configured as a RAID device. In my case, I used that interface to create a 7-drive raid 5 system with one hot spare. FreeNAS sees the 3ware card and all 8 drives as a single disk. From the left navigation tools, select the Disks > Management tool then click on the small plus symbol towards the right edge of the screen to add a drive to be managed by FreeNAS. A subsequent screen will appear that allows selection of all the drives seen by FreeNAS. If there is existing data on the selected drive that needs to be saved, there is an option at the bottom of this subsequent screen to allow selection of the preformatted file system. Be sure to use the 'Apply Changes' button or nothing will be modified.

Of note is that HFS or HFS+ are not options, since these are Apple-centric file systems. If the drives are using HFS or HFS+ and the data needs to be preserved, the drives will need to be backed up from a Mac then restored to the NAS after it has been configured.

Once a disk as been set in the Management panel it can then be formatted using the Disks > Format tool. There are also tools for the creation of Software RAID and data Encryption. In my personal experience, the Software RAID was not reliable and I wouldn't recommend its use for any data that isn't being replicated someplace else. After the Format command is initiated on one of the disks created in the Management section, which might take a few minutes to complete, the webGUI should return a window with the super-block data for the drive. At the bottom of that window (hopefully) will be a 'Done!' message. Using the Disks > Mount Points tool the newly formatted drive should be visible. It should be set with a Share Name.

Using the Access > Users and Groups command first create a Group (from the Group tab at the top of the screen). Then add a user. If your network has an LDAP or AD server, you can point the FreeNAS to that device for a list of users.

The last step in the process is to set up the sharing services. From the Services menu, select AFP (assuming the goal is to put the new NAS device on an Mac network). Click the 'Enable' checkbox in the upper right corner. Then assign a name to the server. Set the Authentication as appropriate. Don't forget to click the 'Save and Restart AFP' button.

Test the box by using the Finder > Go > Connect to Server menu and connect to afp://111.111.111.111 (substituting the IP address assigned to the FreeNAS device). If all went well, a standard OS X authentication will appear. The user name created in the webGUI should allow mounting of the shared drive._

Now What?

The beauty of FreeNAS solution is the unlimited potential to customize the device. There is no reason the kernel or installation environment could not be hacked to install almost any tool available in the FreeBSD ports collection. Some examples relevant to the example system would be installation of the 3ware command line tools to administer the RAID device. To be truly fancy, a full version of Apache could be installed and add the 3ware web administration tool. Ideas I've had, but haven't had the time to execute, would be to create an asset browser for image viewing on the network, or adding a subversion control database to the NAS for collaboration and project management.

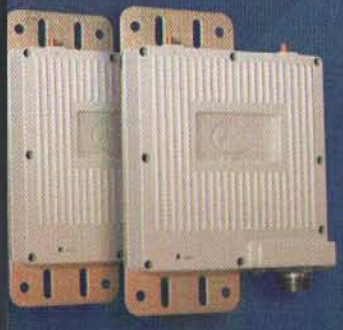
About The Author

VIZf/x is an architectural rendering and animation service organization based in Northern California. Robert Staehle is the principal owner of VIZf/x and is an architect and self-educated sys admin. He can be reached via email at: info@vizfx.com

WE OUTPERFORM OUR COMPETITION

Nobody beats us on...

- **Price**
  Save thousands on multi-location T1 services
- **Performance**
  6 to 30× faster than T1 service
- **Reliability**
  Best-in-class fixed wireless solutions



Just one of many wireless solutions from Trango...

ATLAS 5010 Series™
Wireless Ethernet Bridge

- 45 Mbps of sustained throughput
- Capable of 5.3 and 5.8 GHz
- Point-to-point, OFDM
- Up to 40-mile range
- Adaptable Rate Modulation

Wireless Connectivity IS Powered by Trango

Trango Broadband Wireless
A division of Trango Systems, Inc.
www.trangobroadband.com
Phone: +1 (858) 653-3900

trangobroadband® WIRELESS

Exploring Leopard with DTrace

How to use DTrace for debugging and exploration

by Greg Miller

Introduction

Software is abstract and non-tactile by its very nature. It can be difficult to see what it is doing and why it may be misbehaving. To get a better view of software, we often use tools like `gdb`, `leaks`, `lsof`, and `sc_usage`, just to name a few. We even still use "caveman debugging" techniques like recompiling the code with additional print statements.

A few years back, Sun Microsystems developed DTrace: a new and innovative way to trace running software on live systems. DTrace enables developers and administrators to "see" what their code, and others' code, is doing in a flexible and dynamic way. With the release of Leopard, Apple has brought DTrace to Mac OS X.

This article will begin with a crash course in DTrace. If you're already a seasoned DTrace veteran, feel free to skip that section. We will then move on to some examples of how to use DTrace by exploring our Leopard systems and discovering what makes them purr.

A crash course in DTrace

DTrace is a software tracing facility that can dynamically instrument code by modifying a program after it gets loaded into memory. DTrace can be used on production systems with optimized binaries, without ever having to restart the application – let alone, recompile it! Moreover, DTrace is not limited to tracing user-space applications like most other tracing tools, such as `ktrace`, `strace`, and `truss`. Parts of the system, such as the kernel itself, that were previously off-limits to runtime inspection are now fair game. And, since DTrace instruments code dynamically at runtime, it has zero overhead when not in use.

That said, DTrace will not, and should not, replace all of your other tools. You'll still want to use Shark, Sampler, ObjectAlloc, leaks, Instruments, etc. where they make sense. And let's not forget about good ol' fashion thinking. DTrace is not magic; it's just another tool (albeit a powerful tool) in your toolbox.

DTrace is often used to help answer questions about software, such as "Is function `foo` ever being called, and if so, by whom?", and "How much time is my code spending in the `pwrite` system call?". However, you must know what to ask. If you are tracing your own software, you probably have a good understanding of how it's supposed to work, so coming up with the right questions might not be too difficult. However, if you're using DTrace to explore someone else's software, it may be more difficult to ask the right questions. But never fear; we'll see later that there are some very common questions that are generally good jumping off points. As in life, the answer to one question often begets another. Follow your nose! Using DTrace is very much like surfing the web: your questions about the system are hyperlinks that when clicked will take you to another page full of new links/questions.

You interact with DTrace by writing small programs in the D programming language. These D programs can be saved in text files and run like shell scripts, or they can stretched out right on the command line for quick, ad-hoc use (or if you simply want to impress your friends). An example D script that totals all the system calls made for each process on the system is shown in Listing 1.

Listing 1: syscalls_per_proc.d

Totals up all the system calls made for each process

```
syscall:::entry
{
    @[execname] = count();
```

```
}
```

When run for about 5 seconds on my laptop I got the following output:

```
$ sudo dtrace -s syscalls_per_proc.d
dtrace: script 'syscalls_per_proc.d' matched 427 probes
^C
  Quicksilver                   1
  Finder                        2
  Pages                         2
  DirectoryServic               3
  fseventsd                     3
  mds                           6
  ntpd                         21
  WindowServer                 22
  mDNSResponder                24
  dtrace                       38
  Terminal                     85
```

We could have also specified the D script on the command line as follows:

```
$ sudo dtrace -n 'syscall:::entry {
@[execname] = count() }'
```

Key Concepts and the D Programming Language

The D programming language has syntax very similar to C's and should be very easy for C programmers to use. The D program structure, however, is more akin to AWK's and is made up of one or more clauses of the following form.

```
probe descriptions
/ predicate /
{
    action statements
}
```

One of the key concepts in DTrace is that of a **probe**, which identifies a point (or points) of interest in the kernel or in a user process. Probes are identified by their probe description, which is either a unique integer ID, or more commonly, a 4-tuple written as **provider:module:function:name**.

Fields in a probe description may use shell-style globbing, and omitted fields are assumed to match everything. For example, the probe description `syscall::read:entry` identifies the beginning of the `read` system call by naming the `syscall` provider, any module, the `read` function, and the probename `entry`. The probe description `syscall::*read:entry`, however, identifies both the `read` and `pread` system calls. You can see a list of many (but not all!) of the probes on your system by running `dtrace -l`.

```
$ sudo dtrace -l | wc -l
   40808
```

Each probe may be associated with an optional block of action statements that will be evaluated when the probe fires. A probe may also have an optional predicate that must evaluate to true before the probe's action statements will be called. For example, the following will use the built-in variable `execname` and print the name of each process that calls `read`.

```
syscall::read:entry
{
    printf("%s\n", execname);
}
```

And the following will only print the names of processes that are reading from their standard input (file descriptor 0):

```
syscall::read:entry
/arg0 == 0/
{
    printf("%s\n", execname);
}
```

It is not always insightful to see a separate line of output each time a probe fires. Instead, we may be interested in looking at the data in aggregate form. For example, we could find the total number of bytes allocated by malloc by setting a probe at the entry to malloc, printing out the size argument (`arg0`), then post-processing the data to sum it up. That would work. But DTrace makes this much easier by using aggregating functions and data structures called aggregates. In this way, I could see that Safari mallocs about 2.4MB on my machine when loading my website's homepage.

```
$ sudo dtrace -n 'pid147::malloc:entry {
@total = sum(arg0) }'
dtrace: description 'pid147::malloc:entry ' matched 2 probes
^C
             2414468
```

Here's how that DTrace script (i.e., the argument to `-n`) breaks down:

- The full "probe description" is `pid147::malloc:entry`, which uses the `pid` provider to set a probe at the entry of the `malloc` function in process ID 147 (Safari)
- We do not specify a predicate, so the action statements are always executed when the probe fires
- •The only action statement in this example is the use of the aggregating function `sum`, which totals the argument passed to malloc, and stores the result in the aggregate data structure named `@total`
- Nothing is displayed until we hit ctrl-c; at which point `@total` is displayed
- A large part of writing useful D scripts is knowing what built-in variables and functions are available. The following are a few of the more useful built-in D variables:
- `arg0`, ..., `arg9` – The first 10 arguments to the function matching the probe
- `execname` – The name of the currently executing process
- `pid` – The process ID of the currently executing process
- `ppid` – The parent process ID of the currently executing process
- `probefunc` – The function name of the current probe description

And here are some of the more frequently used data recording functions:

- `trace()` – Evaluates its argument and outputs the result
- `printf()` – Like trace(), but takes a C-style format string
- `stack()` – Records a kernel stack trace

- `ustack()` – Records a user stack trace

DTrace Architecture

Users generally interact with DTrace through the `dtrace(1)` command, which is a generic front-end to the DTrace facility (an alternative front-end is Instruments, which we will not cover in this article). D programs get compiled in user-space and sent to the DTrace virtual machine in the kernel for execution. In other words, the D scripts you write are sent into the kernel and run inside the kernel's address space. They do not run in the `dtrace` process, nor do they run in the target process you are trying to instrument. This is important, and it is the reason we need DTrace functions like `copyin` and `copyinstr`, which are functions that copy data from user space into the kernel's address space.

DTrace providers live in the kernel and can be thought of as plugins to the in-kernel DTrace framework. They are responsible for creating probes by instrumenting various parts of the system. The probes made available by a given provider can be listed using the `dtrace` command. For example, the command `dtrace -l -n proc:::` will list all the probes available from the `proc` provider. The following are some of the providers available on Leopard and what they instrument.

- `syscall` – System calls in the kernel
- `fbt` (Function Boundary Tracing) – Functions in the kernel
- `proc` – Functions related to the process life cycle
- `mach_trap` – Mach traps in the kernel
- `pid` – C functions (or individual instructions) in user space
- `objc` – Objective-C objects in user space

Figure 1 shows the relationship between some of the different components of DTrace.

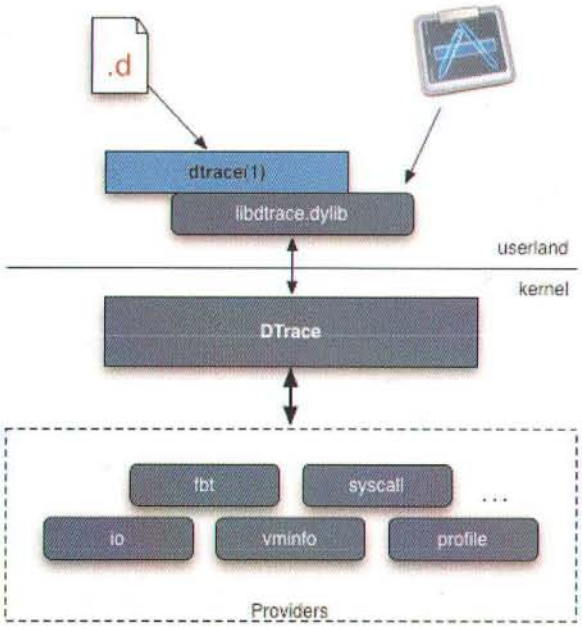


Figure 1. DTrace architecture

Exploring Leopard

Let us now look at a few examples of using DTrace to explore our Leopard system. Note that in order to minimize line wrapping we use `/dev/stdin` as the argument to `dtrace -s`, we type our D scripts right into the standard input, then close standard input by typing ctrl-d (^D).

Tracing Objective-C messages

Tracing system calls and other low-level functions can be fun and insightful, but it may also be too low-level for some situations. Many of the great application frameworks on OS X are written in Objective-C, and it would be nice to trace them at that level. Apple apparently agreed, and they equipped DTrace with a provider for tracing Objective-C messages.

The `objc` provider is very much like the `pid` provider, in that the provider name must include the process ID of the target process. The `objc` provider exposes Objective-C class names as probe modules, and selector names as probe functions. For example, the probe description `objc123:NSView:-isFlipped:entry` would match the entry to the `isFlipped` instance method on the `NSView` class in process 123. Let's try this out by watching what Safari does when it loads www.unixjunkie.net:

```
$ sudo dtrace -q -s /dev/stdin
objc3447:::entry
{
   printf("%s %s\n", probemod, probefunc);
}
^D
dtrace: script '/dev/stdin' matched 66888 probes
(... TONS of output omitted ...)
NSConcreteNotification -recycle
NSObject -retainCount
NSCFString -release
NSObject -release
NSGarbageCollector +defaultCollector
NSLock -lock
NSThread +currentThread
NSObject -hash
NSCFArray -countByEnumeratingWithState:objects:count:
NSLock -unlock
(...)
^C
```

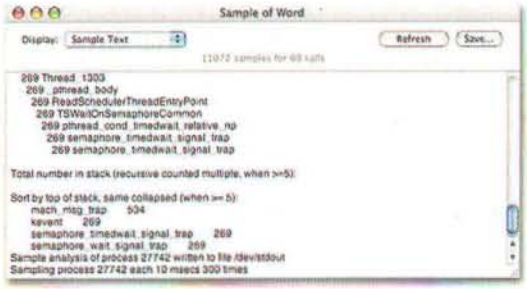
That's cool, but it's a TON of information. With a tool as powerful as DTrace, you can quickly find yourself with more data than you can grok. In this case, we could lessen the output by sharpening our probe description to match only what we're really interested in. For example, if we were only interested in methods dealing with URL handling, we could use the probe description `objc3447:NSURL*::entry`. This quickly cuts the 66,000+ probes down to a manageable 500.

Another way to conquer this mountain of information is with an aggregating function. For example, let's say we're now interested in how long these Objective-C messages take to complete. We could figure this out using the following D script.

Listing 2: objc_msg_times.d

This is a D script to quantize the running time of Objective-

Does



+

= questions?

Are you routinely looking for answers?

Imagine a whole year of answers.

MacTech Magazine is already read every month by tens of thousands of readers. Readers that represent the very heart and soul of the Mac community. Join the crowd and sign up today!

For a special one year subscription, visit:

store.mactech.com

MACTECH®

Toll Free 877-MACTECH, Outside US/Canada: 805-494-9797

C messages. We use the `timestamp` built-in D variable to record the entry time to the Objective-C method in a thread-local variable. Upon the method's return, we quantize the difference between the current `timestamp` and the start time. `$target` is a special D variable that evaluates to the PID of the process under inspection.

```
objc$target:::entry
{
  self->start = timestamp;
}

objc$target:::return
/self->start/
{
  @ = quantize(timestamp - self->start);
  self->start = 0;
}
```

```
$ sudo dtrace -s objc_msg_times.d -p 3447
dtrace: script 'objc_msg_times.d' matched 136042 probes
^C

           value  ------------- Distribution ---------  count
            1024 |                                         0
            2048 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  526082
            4096 |@@@@@@@@@@                               188440
            8192 |@@@@                                     84940
           16384 |@                                        22815
           32768 |                                         6999
           65536 |                                         464
          131072 |                                         82
          262144 |                                         33
          524288 |                                         26
```

```
           value  ------------- Distribution ---------  count
         1048576 |                                         13
         2097152 |                                         9
         4194304 |                                         3
         8388608 |                                         5
        16777216 |                                         0
```

This script introduces a couple new things: thread-local variables and the `quantize` function. D allows you to save variables in thread-local storage by using the `self->` syntax. These variables will be available in other clauses that fire on the same thread. We also use the built-in `quantize` aggregating function to build a power-of-two frequency distribution of the Objective-C messaging times; this can be an incredibly powerful way to interpret the data collected by DTrace. We see here that most of the Objective-C messages completed within 2048–4096 nanoseconds.

File activity

It can be enlightening to see which files are accessed on a system. For example, you may see that Foo.app is frequently writing to some file, or maybe that Bar.app is calling `stat(2)` on a log file every 10ms. This information can help you debug your own programs, or perhaps better understand the system in general. Below we use a small D script to print out the name of each file as it's opened.

```
$ sudo dtrace -s /dev/stdin
syscall::open*:entry
{
   printf("%s %s", execname,
```

```
copyinstr(arg0));
}
^D
dtrace: script '/dev/stdin' matched 3 probes
CPU     ID                    FUNCTION:NAME
  0  17584                      open:entry Finder /.vol/234881026/562669
  0  17584                      open:entry Finder /.vol/234881026/562669
  1  17584                      open:entry iChatAgent /Users/jgm/Library/Caches/...
  0  17584                      open:entry iChatAgent /Users/jgm/Library/Caches/...
  1  17584                      open:entry iChat
/System/Library/PrivateFrameworks/...
^C
```

This script sets a probe at the entry to all system calls having names beginning with "open". DTrace tells us that our probe description matched three probes. They are: `open`, `open_extended`, and `open_nocancel`. Our action statement prints out the name of the process (`execname`) that caused the probe to fire, and the first argument (`arg0`) to the function that matched the probe. Notice that we need to use the `copyinstr` function here rather than just printing `arg0` directly. This is because D scripts execute in the kernel's address space, but the pathname argument to open is stored in user space. We could also modify our D script so that it shows us which files are accessed most often, as follows.

```
$ sudo dtrace -s /dev/stdin
syscall::open*:entry
{
   @[copyinstr(arg0)] = count();
}
^D
dtrace: script '/dev/stdin' matched 3 probes
^C
/Library/Managed Preferences/com.apple.Terminal.plist 1
/Library/Preferences/com.apple.Terminal.plist          1
/Users/jgm/Library/Caches/com.apple.iChat/Pictures/...  1
.
   2
/.vol/234881026/562669
   2
/Users/jgm/Library/Preferences/com.apple.Terminal.plist  3
```

I am a little surprised to see Finder opening files in the `/.vol` directory. Volfs is a separate file system that is used to support the Carbon File Manager atop the BSD file system. It was used on earlier versions of Mac OS X, but it was removed in Leopard.

```
$ sw_vers -productVersion
10.5
$ df -k
Filesystem    1024-blocks      Used Available Capacity  Mounted on
/dev/disk0s2     81732372 38929756  42546616      48%    /
devfs                112      112         0     100%    /dev
fdesc                  1        1         0     100%    /dev
map -hosts             0        0         0     100%    /net
map auto_home          0        0         0     100%    /home
$ ls -al /.vol
total 0
drwxr-xr-x@  2 root  wheel    68 Jul 30 22:08 ./
drwxrwxr-t  31 root  admin  1122 Sep 16 10:22 ../
```

I do not have a volfs file system, but apparently, accessing volfs paths still works.

```
$ ls -l /.vol/234881026/562669
total 3512 .
(... output snipped for brevity ...)
-rw-r--r--   1 jgm  staff        0 Sep 17 11:50
I_am_on_your_desktop.txt
```

Apparently, `/.vol/234881026/562669` refers to my Desktop. We can see which functions are using these paths by looking at the user stack trace when they are opened.

```
$ sudo dtrace -s /dev/stdin
syscall::open*:entry
/copyinstr(arg0) ==
"/.vol/234881026/562669"/
{
   ustack();
}
^D
dtrace: script '/dev/stdin' matched 3 probes
CPU     ID                    FUNCTION:NAME
  1  17584                      open:entry
    libSystem.B.dylib`open$UNIX2003+0xa
    CarbonCore`PBOpenIteratorSync+0x203
    CarbonCore`FSOpenIterator+0x1d

DesktopServicesPriv`THFSPlusIterator::First(THFSPlusRef&)+0x6f

DesktopServicesPriv`THFSPlusIterator::Next(THFSPlusRef&)+0x26
    DesktopServicesPriv`THFSPlusSynchronizer::...+0xb5
    DesktopServicesPriv`TNode::SynchronizeChildren(bool)+0x44
    DesktopServicesPriv`TNode::ReconcileChildren(bool,
bool)+0x63
    DesktopServicesPriv`TNode::HandleSync(bool, bool, bool,
bool)+0x1b1
    DesktopServicesPriv`TNodeSyncTask::...+0xda
    DesktopServicesPriv`TNodeSyncTask::...+0x11f

DesktopServicesPriv`TNodeSyncTask::SyncTaskProc(void*)+0x98
    CarbonCore`PrivateMPEntryPoint+0x38
    libSystem.B.dylib`_pthread_start+0x141
    libSystem.B.dylib`thread_start+0x22
^C
```

Looking at this stack, we can see that it is indeed the Carbon File Manager APIs that are using these volfs paths. Just as we expected. So, even though volfs no longer exists as a file system in Leopard, its main functionality still exists in the kernel to support the Carbon File Manager. Thanks DTrace!

Hard linking directories

One addition in Leopard that has the potential to send shivers up spines, is the addition of directory hard linking. This functionality was added to enable Time Machine to backup large directory structures of unchanged data without wasting space. The canonical argument against hard linked directories is that they can cause cycles in the directory tree. However, Apple avoided this problem by placing some restrictions on directory hard linking.

The only problem I currently see with directory hard links is that I can't get them to work.

```
$ mkdir Dir1
$ ln Dir1 Dir2
ln: Dir1: Is a directory
$ sudo ln Dir1 Dir2
ln: Dir1: Is a directory
```

Perhaps we can use DTrace to figure out what's going on. Let's start like we normally do by looking at all the system calls made by `ln`. We will use dtrace's `-c` option to run and trace the command in question. As we've already seen, the PID of the command is made available to our D script through the `$target` macro variable.

```
$ sudo dtrace -s /dev/stdin -c "ln Dir1
Dir2"
syscall::::entry
/pid == $target/
{}
^D
dtrace: script '/dev/stdin' matched 427 probes
ln: Dir1: Is a directory
dtrace: pid 4389 has exited
CPU     ID                    FUNCTION:NAME
  1  17950                       stat:entry
  1  18368            write_nocancel:entry
  1  18368            write_nocancel:entry
  1  18368            write_nocancel:entry
  1  18368            write_nocancel:entry
  1  17576                       exit:entry
```

This is interesting because we don't see any calls to `link`, which is the system call ultimately responsible for creating the hard link. Let's see if we can get a better view of what `ln` is doing, by using the D script in listing 3.

Listing 3: ln.d

Uses the pid provider to trace all function calls in libSystem. To help limit the output, we only look at user stacks with a depth less than 6.

```
pid$target:libSystem*::entry,
pid$target:libSystem*::return
/ustackdepth < 6/
{}
```

And we will run it like this:

```
$ sudo dtrace -F -s ln.d -c "ln Dir1
Dir2"
dtrace: script 'ln.d' matched 8679 probes
ln: Dir1: Is a directory
dtrace: pid 6171 has exited
CPU FUNCTION
  1  <- __cxa_atexit
  1  -> rindex
  1  <- rindex
  1  -> getopt$UNIX2003
  1  <- getopt$UNIX2003
  1  -> stat
  1    -> _sysenter_trap
  1      -> __error
  1      <- __error
  1      -> warn
  1      <- __error
  1    <- vwarnc
  1  <- fprintf
  1  <- warn
  1  -> exit
  1    -> __cxa_finalize
  1    <- __cxa_finalize
  1    -> _cleanup
  1    <- _fwalk
  1  <- _cleanup
  1  <- exit
  1  -> _exit
  1    -> _sysenter_trap
```

This script traces the function calls in libSystem, and it uses the `-F` (flow indent) option to visually indicate when functions are

entered and return. This output shows that `stat` is called, followed by functions to print the error message. This indicates that `ln` itself is detecting that the first path is a directory and is giving us the error without ever calling `link`. To get past this stumbling block, we will write our own simple C program that calls `link` directly.

Listing 4: hlink.c

Given two file name arguments, creates a hard link from the first to the second.

```
#include <unistd.h>
#include <stdio.h>
int
main(int argc, char *argv[])
{
    if (argc != 3)
        return 1;

    int ret = link(argv[1], argv[2]);
    if (ret != 0)
        perror("link");

    return ret;
}
```

```
$ gcc -o hlink hlink.c -Wall
$ ./hlink Dir1 Dir2
link: Operation not permitted
$ sudo ./hlink Dir1 Dir2
link: Operation not permitted
```

OK, it still didn't work, but at least we know that `link` is definitely being called. We could use DTrace's `fbt` provider to dig further down in the kernel and perhaps see why we're getting this new error, but let's not forget what we learned above: DTrace does not replace all of our other tools. It takes time to write D scripts, so maybe spending a few more seconds just thinking about the problem is a better solution. In our case we know that hard links do indeed work because Time Machine uses them. So, perhaps the problem is that we're trying to create the hard link in the same directory as the original. Let's try creating the hard link to a directory with a different parent.

```
$ mkdir NewDir
$ ./hlink Dir1 NewDir/Dir2
$ ls -lid Dir1 NewDir/Dir2
1140587 drwxr-xr-x  2 jgm  staff  68 Sep 17 18:08 Dir1/
1140587 drwxr-xr-x  2 jgm  staff  68 Sep 17 18:08
NewDir/Dir2/
```

Cool! It worked! Dir2 is a hard link to Dir1 and I don't even feel dirty (nor did I need to be root). We can verify that the hard link worked by looking at the directories' inode numbers (the first column in our `ls` output).

Conclusion

As we've seen here, DTrace is a very powerful and flexible tool, but it is just that – a tool. And it's best used in the hands of a skilled craftsman. The OpenSolaris DTrace community provides a collection of useful D scripts, under the name DTraceToolkit. Thankfully, the code-smiths at Apple have ported many of these scripts to Leopard. You can get a list of the more than 40 available DTrace scripts by running `man -k dtrace`. There are also examples of their use in `/usr/share/examples/DTTk`. One of the more useful D scripts is called `dtruss`, which most users will embrace as a replacement for `ktrace`. Some of the DTrace examples in this article could have been replaced by simpler `dtruss` invocations, but then what fun is that?! We wanted to play with DTrace.

We've just barely scratched the surface of the tip of the DTrace iceberg. We didn't get to touch on the more exotic topics like DTrace's "destructive" actions, the fasttrap provider, instrumenting individual instructions, or accessing external variables using the scoping operator. The technology behind DTrace is impressive, and it is open source so you can see what is happening behind the scenes. Perhaps Apple will provide some DTrace documentation soon, but in the meantime you can find a lot of documentation on Sun's website at http://www.sun.com/bigadmin/content/dtrace, on the OpenSolaris website at http://www.opensolaris.org/os/community/dtrace/, or better yet, just Google it!

About The Author

Greg Miller is a software engineer on the Macintosh Engineering team at Google. He is a bit of a Unix "junkie", but has a passion for many areas of computer science, especially operating system internals. You can reach him through his personal website at http://www.unixjunkie.net.

Easily capture video from any VGA or DVI device

Discover how to grab raw video frames and stream them to your preferred capture utility

By Joe Froehlich

How did you do that?

At one point or another, you've most certainly used the infamous [Command] [Shift] [4] key sequence to take a screen shot of an application window or a dialog box that reports a particular error. And, if you prepare training or technical documentation on a regular basis, you've probably graduated to a more sophisticated screen capture utility like Ambrosia Software's SnapzPro X.

But, how do you capture the screen that appears when you launch the Mac OS X Startup Manager, shown in Figure 1, for example? Furthermore, what if you need to record a movie of the entire boot sequence for troubleshooting or training purposes?

Figure 1: Mac OS X Startup Manager

You can't use the tools we've just described because the operating system isn't loaded yet and there's no way to launch a capture utility. In this article, we'll show you how to capture not only static images like Startup Manager, but entire video streams from any VESA-compatible VGA or DVI source, including industrial, scientific, and forensics equipment.

Discover the magic

To acquire images and data streams like those we're discussing here, you typically need to use a frame grabber that captures the raw data generated at the output of a VGA or DVI source, regardless of whether the source is a computer or some other type of equipment. Most frame grabbers are available only as internal PCI expansion cards. Unfortunately, this limits the type of equipment you can use to capture source content. On the other hand, Epiphan Systems, Inc. provides an external USB solution, obviating the need to rely on internal expansion cards for this task and providing convenient portability at the same time.

Epiphan's product family, appropriately named *VGA2USB*, includes several distinct devices, including the *LR*, *HR*, and *Pro* models, which differ primarily in the image resolutions and frame rates they support. A *DVI2USB* product is also available to support DVI source equipment. A comparative summary of all product specifications is available via the Frame Grabbers link on Epiphan's website (www.epiphan.com). Figure 2 illustrates the standard version of VGA2USB; the other devices look similar but differ in the color of their respective enclosures.

Figure 2: VGA2USB frame grabber with VGA port (left) and mini-USB port (right)

Plug it in

Prior to connecting the product to your equipment, you need to install the streaming software on a system you want to use as a capture workstation; the source equipment doesn't require any software. The streaming software includes the following items, all included in a single installation package:

- The USB device driver
- The `v2u` capture and diagnostics utility (installed in /usr/bin/)
- The `vdig` QuickTime digitizer (installed in /Library/QuickTime/QuickVGA2USB.component)

To install the software, download the latest version of the installation package from www.epiphan.com/downloads/, unpack it, and run the installer. Once the installation is complete, connect the source equipment to the VGA2USB device, and the VGA2USB device to a USB 2.0 port on your capture workstation, as shown in Figure 3.

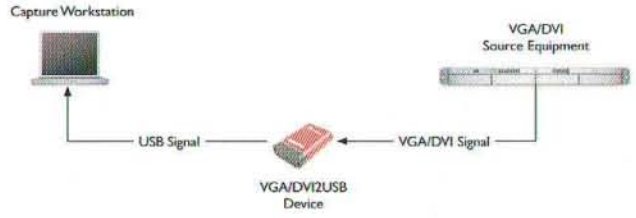


Figure 3: Connecting VGA2USB to your equipment

If desired, you can install either a passive or an active signal splitter to simultaneously display the VGA or DVI output from the source on a secondary display. Doing so helps you monitor the source image during a capture session, for example.

Next, open System Profiler and expand the USB device tree. You'll see the VGA2USB device listed, as shown in Figure 4. Before continuing, make sure the device is connected to a USB 2.0 port.

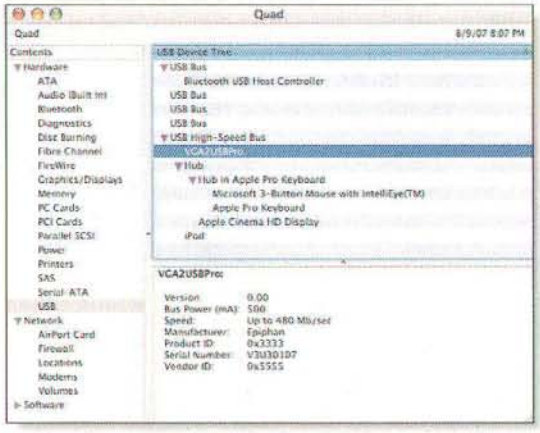


Figure 4: The VGA2USB device as it appears in System Profiler

And finally, open a Terminal session and launch `/usr/bin/v2u`. With no arguments, the `v2u` utility reports the detected signal, as shown in Figure 5; otherwise it reports `no signal detected`. For a complete list of arguments, enter `v2u -h`.

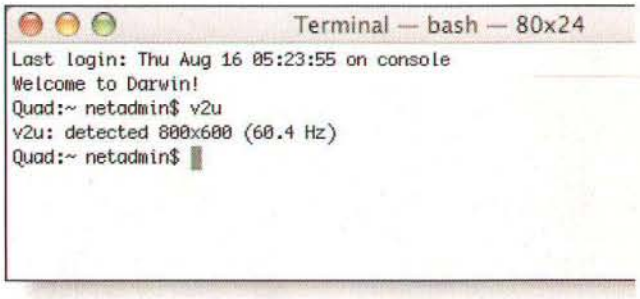


Figure 5: Using the *v2u* diagnostic utility.

Make a movie

While you can use the `v2u` utility to perform diagnostics and capture individual frames, you'll probably want to use an Aqua application, such as QuickTime Pro, to capture video streams. There are several other applications that you can use as well; for example, VGA2USB is certified to work with Adobe Premiere on both Windows and Mac platforms. Other third-party applications include Vara Software's Wirecast and Videocue as well as Boinx Software's iStop Motion.

Because each capture application offers different features and capabilities, it isn't practical to discuss them all here. For the purpose of this article, we'll just give you a quick tour. In Figure 6, for example, the video output of our server machine is being streamed (via VGA2USB Pro) to Wirecast, which is running on our capture workstation. When we click the Record button on the toolbar, Wirecast records a movie of our login session. We can then use QuickTime Player to play back the movie. We could also use Wirecast to broadcast the stream if desired.

Figure 6: Capturing a video stream in Wirecast

Need a Windows solution?

If you wish, you can use a Windows platform to capture your streams as well. Epiphan provides its own Windows capture application, which is installed by default when you install the drivers on a Windows machine. Of course, there are also several third-party applications that work with it as well. Figure 7, for example, shows a PC BIOS screen captured using Epiphan's VGA2USB capture utility.

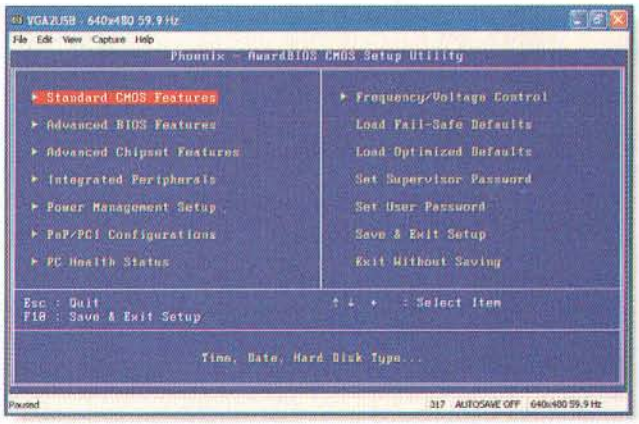


Figure 7: Capturing a PC BIOS screen

"Big deal," you say?

At this point, you may be asking yourself, "Why not just use VNC, ARD, or some other remote monitoring solution?" The answer to this question is that, depending on the screen you need to acquire, these services may only be available much later in the boot process. Furthermore, they require you to alter the source by installing the appropriate software component. You can't use these solutions to capture a screen like Startup Manager or a PC BIOS, much less the video output of a non-computer device.

While our demonstration illustrates a computer technology application, there are other environments in which a frame grabber is not only useful but is actually required, depending on the screen you need to acquire. Figure 8, for example, shows the output of sonogram. Other application environments include:

- Presentations to remote audiences
- Medical diagnostic imaging
- Distance learning
- Scientific device screen acquisition
- Courtroom evidence recording
- Security and surveillance
- Industrial control system monitoring

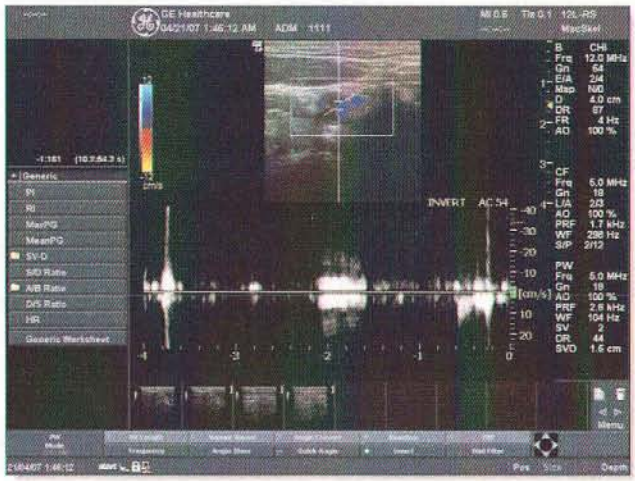


Figure 8: Acquiring a screen from a medical imaging device

In short, if you have a device that generates a VGA or DVI signal and you need to capture its output without altering the source equipment (critical for computer forensics, for example), you need a frame grabber.

Conclusion

In this article, we presented a method for capturing static images as well as video streams from devices that are otherwise inaccessible using traditional screen capture utilities. Hopefully, you've learned something new and, if you have such needs, you'll undoubtedly want to explore solutions like this further.

About The Author

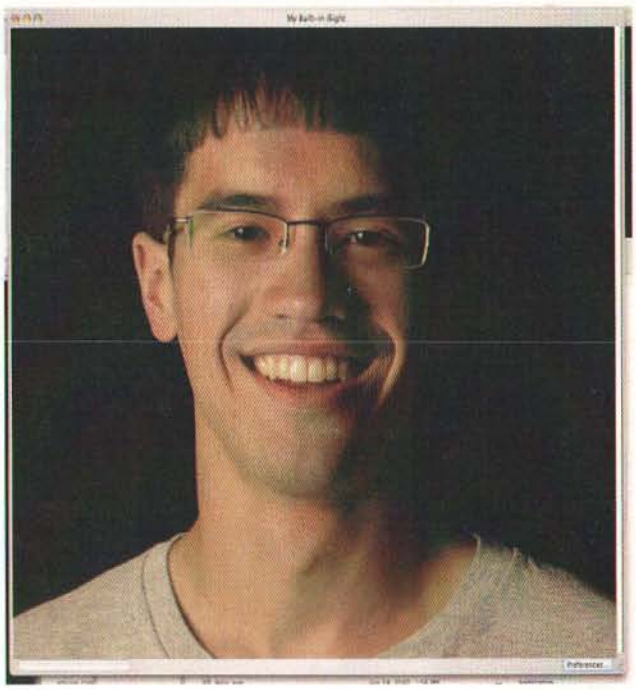
Joe is a technical writer and instructional designer with extensive experience on both Windows and Mac platforms. He's a member of the Apple Consultants Network, the Microsoft Partner Program, and the IEEE Computer Society. You can reach him at froejoe@gmail.com.

Advertiser/Product Index

MACTECH SPOTLIGHT

Kent Sutherland

Independent Developer/Student



http://www.ksuther.com/

What do you do?

I'm a nineteen-year-old, third-year undergraduate computer science student at the Rochester Institute of Technology in Rochester, NY. I work on Chax and other programming projects in my free time. Chax is an InputManager bundle that adds features to iChat. I have also contributed to Adium as part of Google's 2006 Summer of Code.

How long have you been doing what you do?

I initially released Chax in December of 2005, but I've been programming on OS X since my freshmen year of high school. In middle school I dabbled in Mac Toolbox/C and HyperCard, although at that age I wasn't old enough to really understand what I was doing – I just knew that programming was something that I wanted to do.

Your first computer:

The first computer I used was a Mac IIsi that my parents bought when I was about three years old. I was hooked, and I've been a full-blown computer addict and Mac user ever since.

Are you Mac-only, or a multi-platform person?

Through most of high school I was an exclusive Mac user, but I've become more familiar with other platforms in the past few years. I much prefer using and developing on the Mac, but I do own a PC running Windows and Linux. It spends most of its time powered off.

What attracts you to working on the Mac?

A number of different things draw me to the Mac, one of them being the Mac community itself. It's so cool that individuals or small teams of developers can create some really neat programs and get the attention of a large portion of the Mac community. I initially developed Chax for my own use, but it eventually attracted the attention of a bunch of users, blogs, and magazines. Being able to create something and actually make a difference is surprising to me, especially in a fairly short amount of time.

If I could change one thing about Apple/OS X, I'd:

I'd get Apple to put a higher priority on speed. It feels as if Apple doesn't put nearly enough time into optimizing new software. Aperture being a prime example – I remember an early version of Aperture on a quad G5 being virtually unusable. While progressive releases do get faster, it would be nice to see Apple get this right the first time around.

What's the coolest tech thing you've done using OS X?

The coolest thing that I've done in OS X, so far, is Chax. Although the InputManager injection method that Chax uses to load into iChat is one of the simplest ways of getting inside an application, it's cool how much you can do with Objective-C once you're in. Coding inside iChat has also been a great learning experience for me. I've gained a more detailed understanding of Objective-C and better reverse engineering skills as a result.

Outside of OS X, another project that I enjoyed was a computer vision project for a class that performed basic face recognition. There wasn't anything innovative about it, but as I'm still learning more about computer science every day, I found it an interesting subject. As computers get faster and built-in cameras become more common, I'm hoping to see more applications that use these cameras for live input and perform more sophisticated image processing in real time. iChat Theater in Leopard is a fun example, but applications that leverage real-time object or face recognition could be even more useful.

Where can we see a sample of your work?

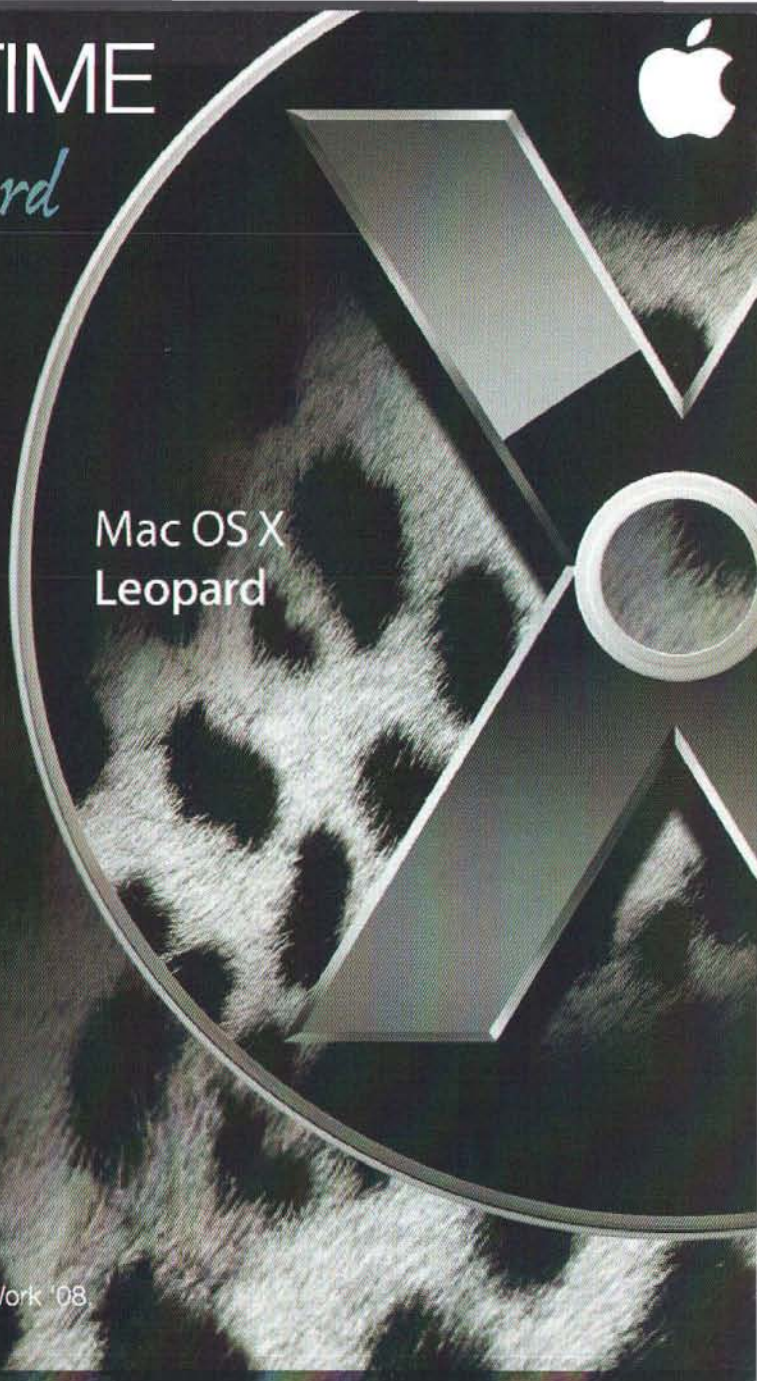
Chax, and its source code, is available for free on my website at http://www.ksuther.com

The next way I'm going to impact IT/OS X/the Mac universe is:

I'm not yet sure what my next big thing is going to be. Continuing with Chax in Leopard and beyond is certainly a possibility, but there's a lot more out there than just working off of Apple's applications. I am definitely going to work on more Mac applications and projects in the future.

T'S ABOUT THAT TIME

Mac OS 10.5 Leopard



Ipgrading to a new operating system doesn't have to ause headaches. Whether you want to upgrade your urrent Mac, or are in the market for a new one, let mall Dog Electronics help you make the most out of 0.5 painlessly!

THE ESSENTIALS

LaCie External Hard Drive

ake advantage of Time Machine in Leopard and back p all of your valuable data onto a separate hard drive. or either desktops or notebooks, we have a wide selec- on of LaCie drives to fit your needs!

Extra RAM

on't bog down your Mac-- make those processors ork for you as well as they possibly can. Our RAM rices are really competitive and we offer free installation ith a Mac purchase!

Life '08 and iWork '08

Ipgrade your current Mac with iLife '08 to utilize all that eopard has to offer, and be sure to try Numbers, a fully ompatible spreadsheet application-- now in iWork '08!

ew Macs come with the latest version of iLife, and a 30-day trial of iWork '08.

Ve here at Small Dog have earned our place in the Apple community ith our famously superior customer service, great prices, and by taying true to the Apple platform.

Ve are here to help you, so if you have questions about Leopard or nything else Mac-related, give us a call or visit us at Smalldog.com!

Loyal MacTech readers:
Get $10 off your next purchase of $75 or more. Enter "mactech10" at checkout.

Coupon good through 12/31/07

Small Dog Electronics

Always By Your Side

www.smalldog.com

800-511-MACS

Apple Specialist

THINGS MADE IN CHINA

COMPRESSED BY

STUFFIT DELUXE

SOFTWARE FOR MAC/PC